AÑO III - NUM. 77

MICROHOBBY

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR
SEMANAL
AÑO III - N.º 77

135 PTS.
Canarias 140 ptas.



UTILIDADES

RESET POR SOFTWARE

¡RECUPERA EL CONTROL SIN PERDER LA INFORMACIÓN!

TOP SECRET

CARGADORES OCULTOS EN LA ZONA DE VARIABLES

INICIACION

INTERCAMBIO DE INFORMACIÓN

BASIC–CÓDIGO MÁQUINA

NUEVO

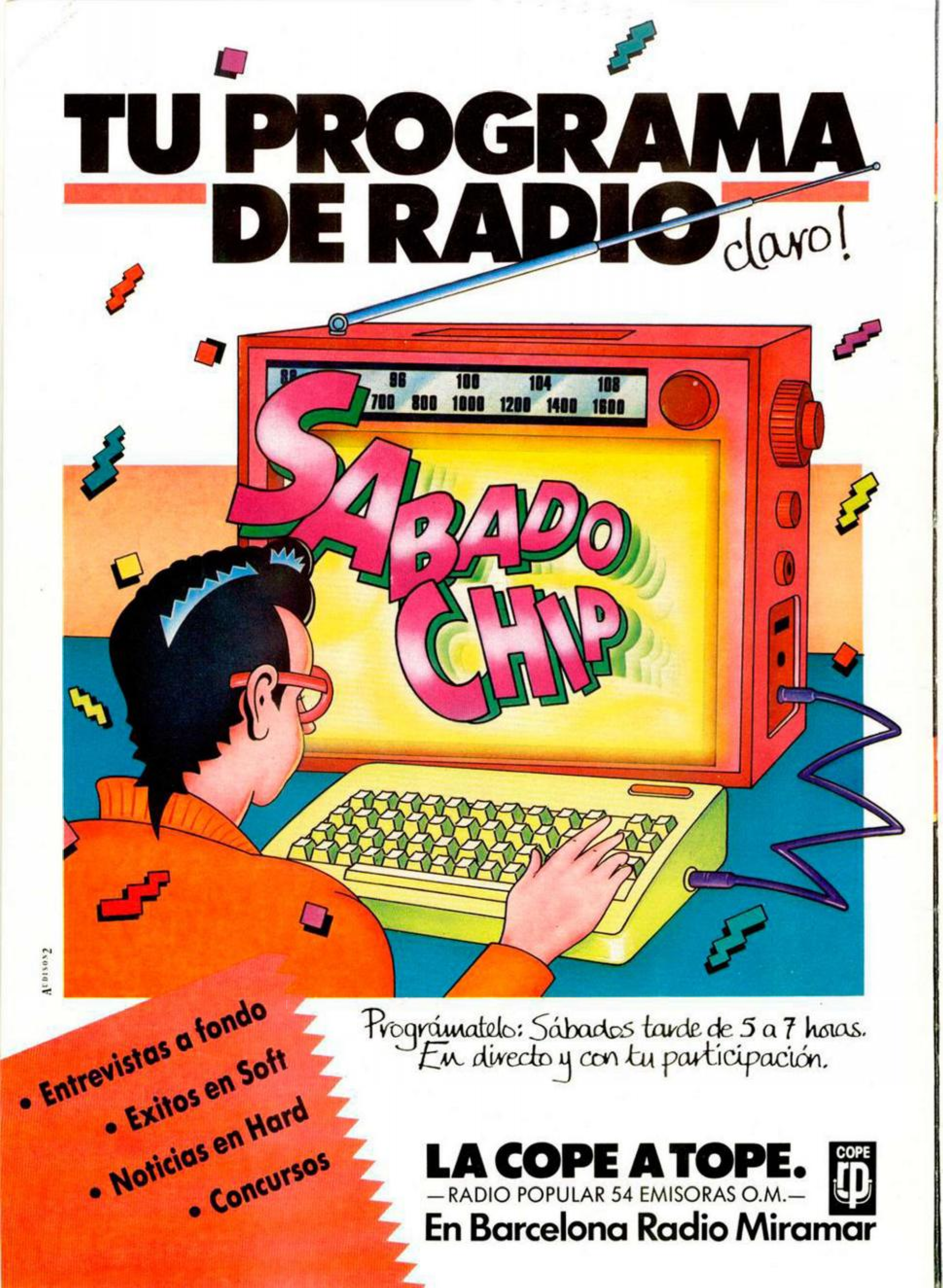
GREEN BERET:

La trepidante aventura de un Boina Verde

MICROMANIA

VIDAS INFINITAS PARA EL MUNDO DE SWEEVO

HOBBY PRESS


TU PROGRAMA
DE RADIO
claro!
96 100 104 108
700 800 1000 1200 1400 1600
SABADO CHIP
Prográmatelo: Sábados tarde de 5 a 7 horas.
En directo y con tu participación.
• Entrevistas a fondo
• Exitos en Soft
• Noticias en Hard
• Concursos
LA COPE A TOPE.
—RADIO POPULAR 54 EMISORAS O.M.—
En Barcelona Radio Miramar
COPE
rp

Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Domingo Gómez
Asesor Editorial
Gabriel Nieto
Redactora Jefe
Africa Pérez Tolosa
Diseño
Rosa María Capitel
Redacción
Amalio Gómez, Pedro Pérez,
Jesús Alonso
Secretaria Redacción
Carmen Santamaría
Colaboradores
Primitivo de Francisco, Rafael Prades,
Miguel Sepúlveda, Sergio Martínez
y J. M. Lazo
Corresponsal en Londres
Alan Heap
Fotografía
Carlos Candel
Portada
José María Ponce
Dibujos
Teo Mójica, F. L. Frontán,
J. M. López Moreno,
J. Igual, J. A. Calvo,
Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.
Presidente
María Andrino
Consejero Delegado
José I. Gómez-Centurión
Jefe de Producción
Carlos Peropadre
Jefe de Publicidad
Concha Gutiérrez
Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76
Secretaria de Dirección
Pilar Aristizábal
Suscripciones
M.ª Rosa González
M.ª del Mar Calzada
Redacción, Administración
y Publicidad
La Granja, 39
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR
Dto. Circulación
Paulino Blanco
Distribución
Coedis, S. A. Valencia, 245
Barcelona
Imprime
Rotedic, S. A. Ctra. de Irún,
km 12,450 (MADRID)
Fotocomposición
Novocomp, S.A.
Nicolás Morales, 38-40
Fotomecánica
Grof
Ezequiel Solana, 16
Depósito Legal
M-36.598-1984
Representante para Argentina,
Chile, Uruguay y Paraguay, Cía.
Americana de Ediciones, S.R.L.
Sud América 1.532. Tel.: 21 24 64.
1209 BUENOS AIRES (Argentina)

Solicitado control
OJD

MICROHOBBY ESTA SEMANA

AÑO III. N.º 77. 6 al 12 de mayo de 1986.
135 ptas. Canarias, Ceuta y Melilla: 130 ptas.
Sobretasa aérea para Canarias: 10 ptas.





Cómo enlazar el C.M. y el Basic del Spectrum. Pág. 22.

MICROHOBBY NUMEROS ATRASADOS

Queremos poner en conocimiento de nuestros lectores que para conseguir números atrasados de MICROHOBBY SEMANAL, no tienen más que escribirnos indicándonos en sus cartas el número deseado y la forma de pago elegida de entre las tres modalidades que explicamos a continuación.
Una vez tramitado esto, recibirá en su casa el número solicitado al precio de 95 ptas. + 6 de IVA hasta el n.º 36, a 125 ptas. + 8 de IVA hasta el n.º 60 y a 135 ptas. desde el n.º 60 en adelante.



FORMAS DE PAGO

- Enviando talón bancario nominativo a Hobby Press, S. A. al apartado de Correos 54062 de Madrid.
- Mediante Giro Postal, indicando número y fecha del mismo.
- Con Tarjeta de Crédito (VISA o MASTER CHARGE), haciendo constar su número y fecha de caducidad.

MICROPANORAMA

Una de las compañías más potentes en la fabricación de ordenadores

COMPAQ LLEGA AL MERCADO ESPAÑOL

El vicepresidente de COMPAQ en EE.UU. y el director general de OTESA, presidieron una conferencia de prensa en la que se celebró la presentación oficial de Compaq en España.

OTESA es la distribuidora exclusiva para España de algunos productos de las más prestigiosas casas mundiales dedicadas a la fabricación de ordenadores personales y equipos electrónicos y de oficina en general como pueden ser Toshiba, Silver Reed, Hewlett Packard, y ahora, COMPAQ, quien ha sido durante tres años seguidos, la compañía con mayor crecimiento mundial en electrónica ocupando la primera posición en el mundo en cuanto a ventas de microordenadores portátiles. Ahora COMPAQ es noticia en España por la presentación de tres modelos de ordenadores portátiles que serán puestos a la venta próximamente en nuestro país.

COMPAQ PORTABLE II: microprocesador 80286 de 16 bits, compatible IBM PC, 2 slots de expansión, monitor en color, 256 kbytes de Ram y una o dos unidades de disco según modelo.

COMPAQ PORTABLE: microprocesador de 16 bits 8088, compatibilidad IBM PC, una o dos unidades de disco con capacidad de 360 K, 3 slots, 128 K de memoria RAM (expandible a 640 K), monitor de fósforo verde.

COMPAQ DESKPRO: microprocesador 8086 de 16 bits, compatible IBM PC, 128 kbytes de RAM (expandible a 640 K), 6 slots de expansión, una o dos unidades de disco de 360 K, monitor en fósforo verde con alta resolución de gráficos y caracteres.

INMINENTE LANZAMIENTO DE PHANTOMAS DE DINAMIC

Dinamic anuncia la inminente aparición de su última creación: **Phantomas**, cuya realización ha sido definitivamente terminada.

Este nuevo programa está en la línea, en cuanto al estilo gráfico y de acción se refiere, a la utilizada en el ya mítico Abu Simbel Profanation, aunque, lógicamente, se han añadido una serie de innovaciones a todos los niveles con las que se ha llegado a alcanzar una gran calidad.



La acción de Phantomas se desarrolla en el planeta clónico Earth-Gamma donde se encuentra escondido un fabuloso tesoro perteneciente al avaro y multimillonario Goldter. Allí, nuestro pequeño Phantomas, un mutante especializado en el robo, saqueo y pillaje, deberá ir superando tres etapas diferentes para lograr hacerse con tan preciado capital. Estas son: la base espacial, en la que encontraremos diferentes tipos de naves como cohetes, helicópteros, etc.; el Palacio de Goldter, el lugar más peligroso, en donde nos encontraremos con toda clase de trampas y obstáculos, y por último, el Mundo Subterráneo, habitado por los antepasados del multimillonario y cuyos restos descansan eternamente encerrados en sus respectivas criptas.



Phantomas va a realizar, de la mano de Dinamic, el robo más grande de la historia espacial. Suponemos que la misión será, como siempre, un nuevo éxito.

AQUI LONDRES

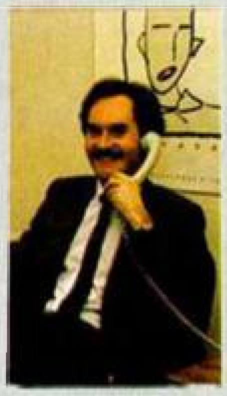
Parece ser que, a pesar de los últimos acontecimientos, **el QL sobrevivirá**, pero quizá de una forma muy distinta. Un consorcio de compañías independientes planea comprar los derechos del QL si Amstrad decide suspender la producción de dicha máquina. En tal caso, el QL probablemente será ampliado a 512 K y tendrá una unidad de discos de 3/2", con lo cual pasará a ser una parte integral del mismo.



Hewson Consultans, ha anunciado el lanzamiento inminente de **«Quazatron»** para el Spectrum. Este juego es una continuación de su último éxito realizado por la casa Paradroid. Quazatron será lanzado a principios de Mayo, y su precio será de 9 libras.

Level ha producido una versión mejorada de **«The Pric of Magik»** para el Spectrum 128 K. Esta versión, que costará 10 libras, aparecerá a finales del mes de abril.

Tras abandonar sus la bores de distribución, **Ultimate** estrena casa distribuidora, U.S. Gold, con su programa de más reciente creación: **Cyberun.** El precio en el mercado de este juego alcanza las 10 libras, cantidad ésta, reservada a los programas de mayor calidad.



U.S. Gold va a sacar una gama de software barato con un precio de3 libras en lugar de 10, con la finalidad de ampliar el software para los ordenadores de 16 Bit (C-64, Atari, 128 K y Amiga).

De nuestro corresponsal en Londres ALAN HEAP

DATAFONO: UN MILLON DE SERVICIOS

La Compañía Telefónica Nacional de España celebró en las instalaciones de la madrileña calle de Recoletos, un acto en el que se celebró la **llamada un millón** efectuada a través de uno de sus más modernos servicios: el Datáfono o teléfono de datos.

Este equipo de abonado, cuyo diseño y fabricación tienen lugar en España con tecnología propia, surge por la necesidad de mejorar y ampliar la red de datos nacionales derivados del uso y servicio de los cajeros automáticos y equipos de servicios de ventas.

Sin embargo, las posibles utilizaciones del Datáfono son muy numerosas y de muy diferente naturaleza. Como aparato telefónico puede realizar todas las operaciones de un teléfono convencional, pero además, ofrece otras nuevas e interesantes funciones como pueden ser: supervisión de línea, informando mediante mensajes de su estado; marcación automática de hasta diez números previamente almacenados; rellamada automática, detención de tonos y recepción por altavoz.

Como equipo de datos, el Datáfono puede comunicarse con un centro de cálculo abonado por la Red Telefónica Conmutada o bien por la Red de Transmisión de Datos (IBERPAC). Los datos de cada transacción se van almacenando en un registro para permitir la depuración y corrección de los mensaje antes de su transmisión. La entrada de dichos datos puede realizarse a través del teclado principal, por el teclado auxiliar o por la lectura de la banda magnética de la tarjeta. La transmisión recepción de los datos se efectúa a una velocidad de 300 b.p.s. y éstos pueden destinarse al visualizador, a la impresora o a ambos.

De entre las características técnicas de su Unidad de Control, cabe destacar los siguientes puntos: microprocesador MC 146805 E2 de bajo consumo y 8 bits; dispone de 4 K octetos de memoria Eprom para sus programas operativos, 1 K octetos de memoria RAM para datos variables y 512 octetos de memoria Eprom para almacenar datos.

Las posibles aplicaciones del Datáfono son múltiples, pero cabría resaltar su gran utilidad como teléfonos en los puntos de venta para el tratamiento de operaciones realizadas con tarjetas de crédito, teléfono de directivo, con posibilidades de acceso a informaciones de control, estadísticas o bancos de datos, como teléfono puramente convencional o como equipo informático en pequeñas oficinas.



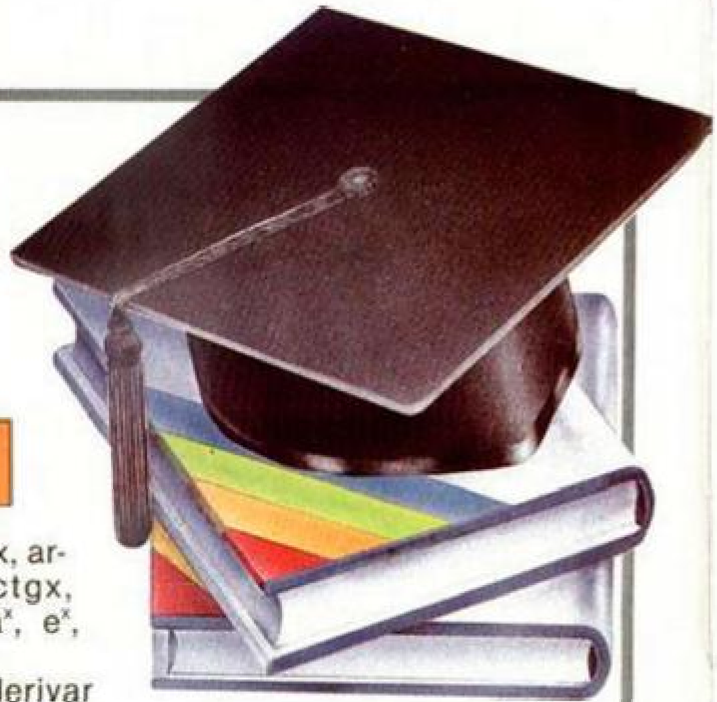
CLASIFICACION	SEMANAS PERM.	TENDENCIA	20 +	SPECTRUM	AMSTRAD	COMMODORE	MSX
1			**RAMBO.** Ocean		●		
2			**KNIGHT LORE.** Ultimate				●
3			**KUNG-FU MASTER.** U.S. GOLD			●	
4			**ALIEN 8.** Ultimate				●
5			**RAMBO.** Ocean	●			
6			**CAMELOT WARRIORS.** Dinamic	●			
7			**AJEDREZ 3D.** Cyrus		●		
8			**SOCCER.** Konami				●
9			**DESERT FOX.** U.S. GOLD			●	
10			**COMMANDO.** Elite	●			
11			**LA LEYENDA DE LAS AMAZONAS.** Silver Time	●			
12			**YIE AR KUNG-FU.** Imagine	●			
13			**THEY SOLD A MILLION.** U.S. GOLD	●			
14			**SIR FRED.** Made in Spain		●		
15			**COMMANDO.** Elite			●	
16			**WORLD SERIES BASKETBALL.** Imagine	●			
17			**SKY FOX.** Electronics Arts		●		
18			**RAMBO.** Ocean			●	
19			**COLOSUS CHESS.** CDS			●	
20			**MATCH-DAY.** Ocean	●			

Con la intención de ofreceros una lista que corresponda fielmente a la realidad del índice de ventas alcanzado por los distintos programas disponibles en el mercado, hemos decidido solicitar la colaboración de los departamentos de informática de los centros de El Corte Inglés de toda España.

Esta semana hemos elaborado una primera lista provisional en la que aparecen una relación fidedigna de los juegos que han alcanzado un mayor índice de ventas en sus versiones para los diferentes ordenadores.

El Corte Inglés

Profesor particular



Arturo LOBO y J.J. LEON

DERIVADAS 2

Ya tenéis aquí la segunda parte del programa de derivadas, que como sabéis, tiene que unirse a la primera para funcionar correctamente. Y, como os prometimos, os vamos a dar unas pequeñas normas que podéis seguir aquéllos que tengáis dificultad especial para derivar. Creemos que con el programa y este resumen podréis conseguir hallar cualquier función derivada, aunque debéis pensar también que la facilidad sólo puede obtenerse con la práctica.

Y aquí vienen esas 10 «lecciones»:

1. La derivada de una constante es cero.
2. La derivada de x es 1.
3. La derivada de ax^b es $abx^{(b-1)}$.
4. La derivada de una suma es la suma de las derivadas.
5. La derivada de un producto uv es la derivada de u por v más la derivada de v por u.
6. La derivada de un cociente u/v es la derivada de u por v menos la derivada de v por u y todo dividido por u^2.
7. Debéis recordar de memoria la derivada de una serie de funciones elementales: senx, cosx, tagx, arcsenx, arccosx, arctgx, logx, Lnx, a^x, e^x, $\sqrt{x}$
8. Para derivar $\sqrt[a]{x}$, considerarlo como $x^{1/a}$ y operar igual que en 3.
9. Regla de la cadena: si queréis derivar algo que no aparezca explícitamente en los puntos anteriores, derivar la expresión más «exterior» y a continuación, poner un signo por, abrir un paréntesis y proseguir derivando la expresión que queda como si la más exterior ya no estuviera (el programa lo muestra más claramente).
10. Derivar una función elevada a otra se hace derivando primero como potencia, luego como exponencial y sumando ambos resultados.

Os aseguramos que si hacéis los ejercicios de derivadas delante de vuestro Spectrum no sólo os resultará más entretenido sino que el aprendizaje visual podrá ahorrarnos muchas horas de trabajo.

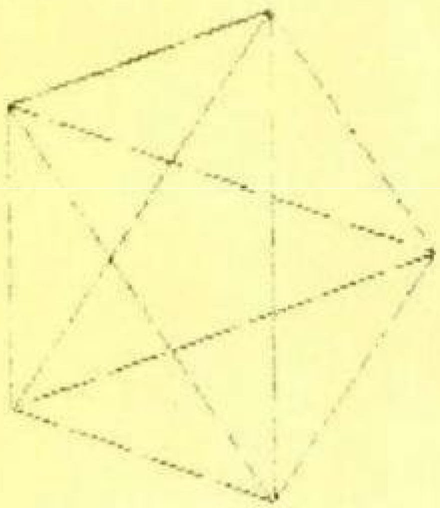
```
8102 LET solo=0: IF a$(2 TO LEN a$)=i$ THEN LET solo=1
8105 GO TO (7930+CODE a$(1))
8108 LET a$="COS "+a$(2 TO LEN a$)+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8109 LET a$="(-SIN "+a$(2 TO LEN a$)+")"+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8110 LET a$="1/COS "+a$(2 TO LEN a$)+"↑2"+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8111 LET a$="1/SQR (1-"+a$(2 TO LEN a$)+"↑2)"+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8112 LET a$="(-1)/SQR (1-"+a$(2 TO LEN a$)+"↑2)"+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8113 LET a$="1/(1+"+a$(2 TO LEN a$)+"↑2)"+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8114 LET a$="1/"+a$(2 TO LEN a$)+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8115 LET a$=a$+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8117 LET a$="1/2/"+a$+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8119 LET a$=CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8130 IF solo=1 THEN LET a$=a$(1 TO (LEN a$-LEN i$-4))
8131 RETURN
8200 LET c$=a$(b(2)+1 TO b(3)-1): FOR i=3 TO no: LET c$=c$+"*"+a$(b(i)+1 TO b(i+1)-1): NEXT i: IF no>=3 THEN LET c$="("+c$+")"
8210 LET d$=a$(1 TO b(2)-1): GO SUB 8480: LET pvar1=pvar: LET d$=c$: GO SUB 8480: LET pvar2=pvar: IF pvar1+pvar2<>0 THEN LET a$=a$(1 TO b(2)-1)+"↑"+c$+"*(LN "+a$(1 TO b(2)-1)+"*"+CHR$ 0+"("+c$+")"+"+"+c$+"/"+a$(1 TO b(2)-1)+"*"+CHR$ 0+"("+a$(1 TO b(2)-1)+"))": RETURN
8220 IF pvar2<>0 THEN LET a$=a$(1 TO b(2)-1)+"↑"+c$+"*LN "+a$(1 TO b(2)-1)+"*"+CHR$ 0+"("+c$+")": RETURN
8222 DEF FN a(x)=FN l(x)+((x<48)+(x)=58))
8225 LET tra=0: FOR i=1 TO LEN c$: LET tra=tra+FN a(CODE c$(i)): NEXT i: IF tra=0 THEN LET c$=STR$ (VAL c$): IF VAL c$=0 THEN LET a$="0": RETURN
8227 IF c$="1" THEN LET a$=CHR$ 0+"("+a$(1 TO b(2)-1)+")": RETURN
8230 LET a$=c$+"*"+a$(1 TO b(2)-1)+"↑("+c$+"-1)"+"*"+CHR$ 0+"("+a$(1 TO b(2)-1)+")": RETURN
8300 LET c$=""
8310 FOR i=1 TO no: LET d$=a$(b(i)+1 TO b(i+1)-1): GO SUB 8480: IF pvar=0 THEN NEXT i: GO TO 8350
8320 LET d$="": FOR j=1 TO no: IF j<>i THEN LET d$=d$+a$(b(j)+(j=1) TO b(j+1)-1): NEXT j: GO TO 8340
8325 IF a$(b(i)+(i=1))="/" THEN LET sg=45: LET d$=d$+"/"+a$(b(i)+1 TO b(i+1)-1)+"↑2*"+CHR$ 0+"("+a$(b(i)+1 TO b(i+1)-1)+")": NEXT j: GO TO 8340
8330 LET sg=43: LET d$=d$+"*"+CHR$ 0+"("+a$(b(j)+1 TO b(j+1)-1)+")": NEXT j
8340 LET c$=c$+CHR$ sg+d$: NEXT i
8350 IF (c$(1 TO 2)="+*")+(c$(1 TO 2)="-*")=1 THEN LET c$=c$(3 TO LEN c$)
8352 IF c$(1)="+" THEN LET c$=c$(2 TO LEN c$)
8355 IF paren=1 THEN LET a$="("+c$+")": RETURN
8358 LET a$=c$: RETURN
8400 LET d$=a$(b(1)+1 TO b(2)-1): GO SUB 8480: IF pvar<>0 THEN LET c$=CHR$ 0+"("+d$+")": GO TO 8410
8405 LET c$=""
8410 FOR i=2 TO no: LET d$=a$(b(i)+1 TO b(i+1)-1): GO SUB 8480: IF pvar<>0 THEN LET c$=c$+a$(b(i))+CHR$ 0+"("+d$+")"
8415 NEXT i: IF paren=1 THEN LET a$="("+c$+")": RETURN
8420 LET a$=c$: RETURN
8480 IF d$="" THEN LET pvar=0: RETURN
8490 LET pvar=0: LET fin=LEN d$-LEN i$+1: FOR p=1 TO fin: IF d$(p TO (p-1+LEN i$))=i$ THEN IF ((FN l(CODE d$(p-1+(p=1)))=0)+(p=1))*((FN l(CODE d$(p+LEN i$-(p=fin)))=0)+(p=fin))=1 THEN LET pvar=p: RETURN
8492 NEXT p: RETURN
9550 BEEP .5,50: PRINT AT 10,0;: FOR k=1 TO ax: PRINT a$(a(k,2)+1 TO a(k+1,1)-1);: IF k<>ax THEN PRINT INVERSE 1;a$(a(k+1,1) TO a(k+1,2));: NEXT k
9555 GO TO 9590
9560 PRINT AT 13,0; FLASH 1;a$(1); FLASH 0;a$(2 TO LEN a$);
9565 GO TO 9590
9570 PRINT AT 13,0;: FOR k=1 TO no: PRINT a$(b(k)+1 TO b(k+1)-1);: IF k<>no THEN PRINT FLASH 1;a$(b(k+1));: NEXT k
9575 GO TO 9590
9580 PRINT AT 10,0;: FOR g=1 TO 4*32: PRINT " ";: NEXT g: RETURN
9590 BEEP .1,20: BEEP .1,0: PRINT AT 21,15;"C- Continuar"
9592 LET j$=INKEY$: IF j$<>"" THEN PRINT AT 21,0;"                ": RETURN
9594 GO TO 9592
9600 LET vai=0: LET par=0: LET vax=0: LET r$=""
9605 INPUT "DIFICULTAD de 1 a 10 ";dif
9610 LET dif=1.5*dif: GO TO 9670
9640 IF FN r(2)>1.5+vax+(dif<3) THEN LET r$=r$+"x": LET vax=1: GO TO 9642
9641 LET r$=r$+CHR$ (49+FN r(8))
9643 LET rx=FN r(10): IF (par>0)*((rx<3+2*(dif/LEN r$) OR (dif*2<rx)) THEN GO TO 9650
9644 GO TO 9660+20*(par=0)*(vax OR (dif<=2))*((dif/LEN r$<rx/6)
9650 LET par=par-1: LET r$=r$+")": LET rx=FN r(10)
9652 IF par=0 THEN GO TO 9660+20*(vax OR (dif<=2))*(2*FN r((dif/LEN r$)<3)
9654 GO TO 9660-10*(FN r((dif/LEN r$)<2)
9660 LET qw=FN r(5): LET qw=42*(qw=1)+43*(qw=2)+45*(qw=3)+47*(qw=4)+94*(qw=5): LET r$=r$+CHR$ qw
9662 GO TO 9670-30*((FN r(10*(dif/LEN r$)<3) OR (FN r(5)>dif))
9670 IF (FN r(12)>3+3*par+4) AND (FN r(3)<=dif) THEN LET par=par+1: LET r$=r$+"(": GO TO 9672
9671 LET r$=r$+CHR$ (177+FN r(8))
9672 IF (FN r(20)>6+(3+par)*((dif/(3+1.5*LEN r$))) OR (dif<FN r(5)) THEN GO TO 9640
9674 GO TO 9670
9685 LET e$=r$: RETURN
9690 IF ense<3 THEN RETURN
9692 PRINT AT 15,0;"Que derivando queda ";a$: GO TO 9750
9700 IF ense<2 THEN RETURN
9703 CLS : PRINT AT 0,0;"Funcion ";g$
9706 PRINT AT 3,0;"Tenemos ";b$; PAPER 7; INVERSE 1;e$(0 TO i);
9710 PRINT e$(i+1 TO LEN e$);
9715 IF ense=2 THEN GO TO 9750
9720 RETURN
9750 BEEP .1,10: PRINT AT 20,20;"De acuerdo ?"
9755 IF INKEY$<>"" THEN PRINT AT 20,20;"            ": RETURN
9760 GO TO 9755
9780 PRINT AT 3,0;"Derivando ";e$: GO TO 9750
```

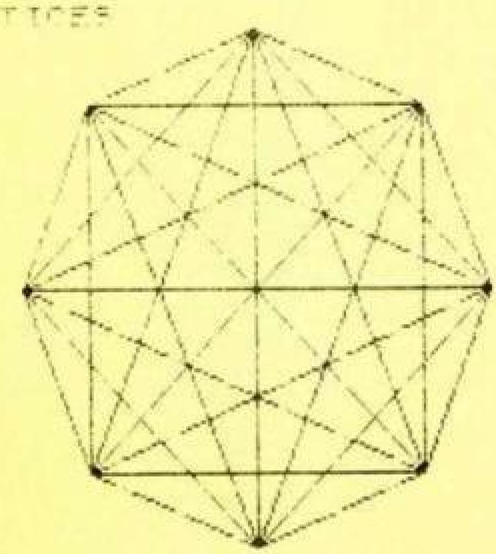
TRUCOS

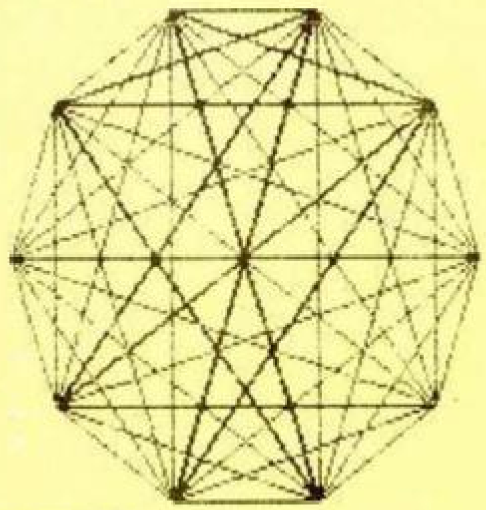
FORMAS GEOMETRICAS

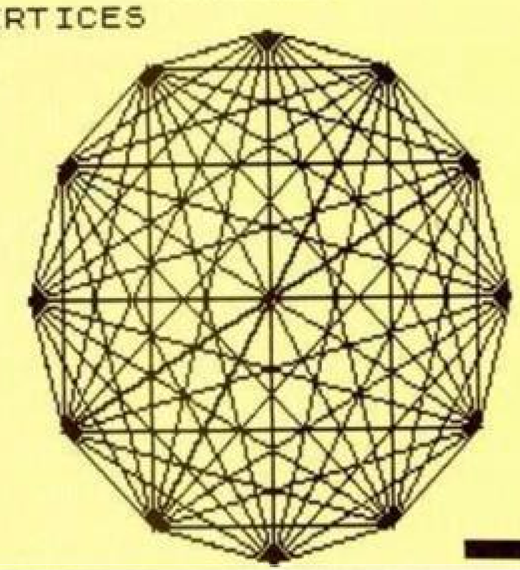
A partir de un número determinado de vértices (de 2 a 20), José Verdú nos muestra que se puede unir cada uno de ellos con todos los restantes logrando así una bonita composición de formas geométricas.

```
 10 FOR n=2 TO 20
 15 PRINT AT 0,0;n;" VERTICES"
 20 LET R=80
 30 LET IA=2*PI/N
 40 LET A=0
 60 FOR U=1 TO N
 70 FOR I=1 TO N-1
 80 LET X1=127+R*COS (A): LET Y
1=87+R*SIN (A)
 90 PLOT X1,Y1
100 LET X2=127+R*COS (A+(IA*I)):
 LET Y2=87+R*SIN (A+(IA*I))
110 DRAW X2-X1,Y2-Y1
120 NEXT I
130 LET A=A+IA
140 NEXT U
150 BEEP .05,20: PAUSE 20: CLS :
 NEXT N
```









LA PIRAMIDE

Siguiendo con los bonitos efectos, Lawrence Humprey nos manda esta rutina para conseguir la pirámide que aquí os mostramos.

```
               A
              BBB
             CCCCC
            DDDDDDD
           EEEEEEEEE
          FFFFFFFFFFF
         GGGGGGGGGGGGG
        HHHHHHHHHHHHHHH
       IIIIIIIIIIIIIIIII
      JJJJJJJJJJJJJJJJJJJ
     KKKKKKKKKKKKKKKKKKKKK
    LLLLLLLLLLLLLLLLLLLLLLL
   MMMMMMMMMMMMMMMMMMMMMMMMM
  NNNNNNNNNNNNNNNNNNNNNNNNNNN
 OOOOOOOOOOOOOOOOOOOOOOOOOOOOO
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
```

```
 6 REM PYRAMIDE
 7 REM LAWRENCE HUMPREY
 8 REM NO HAY UNA MANERA MAS
 9 REM FACIL DE HACER ESTO?
10 LET N=65: LET T=16: LET P=0
15 FOR Y=1 TO 31 STEP 2
20 FOR X=1 TO Y
30 PRINT AT P,T;CHR$ N;
35 LET T=T+1: NEXT X
40 LET T=15-INT Y/2: LET N=N+1
45 LET P=P+1: NEXT Y
```

KRAKATOA

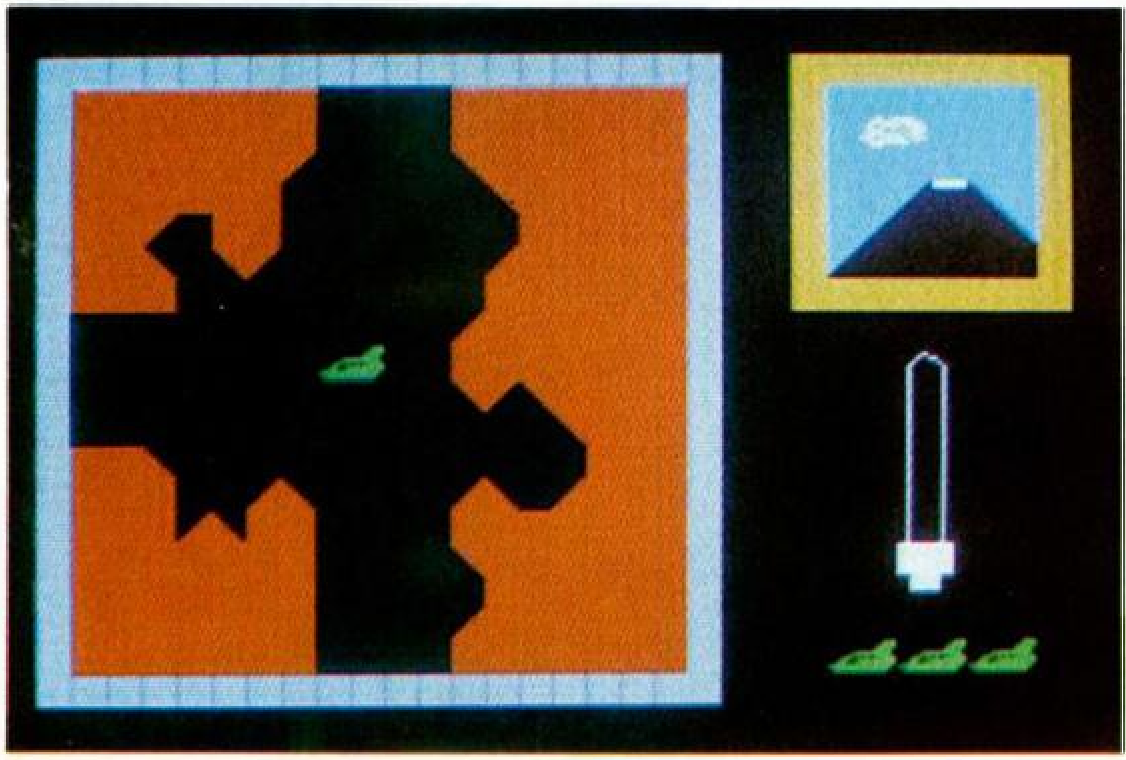
Vicente J. JIMENEZ

SPECTRUM 48 K

La isla de Java está azotada nuevamente por las turbulencias de un volcán histórico por sus erupciones: el Krakatoa.

Una afamada expedición de científicos que habían acudido a la isla para estudiarlo se han visto sorprendidos por la erupción y uno de ellos, el profesor Volcánicus Eruptus, se encuentra en una situación de grave peligro siendo necesaria nuestra pronta ayuda para rescatarle del monstruo de fuego (una denominación del volcán muy común en la zona) antes de que la elevada temperatura haga explotar al Krakatoa.

¿Lo conseguiremos? Para intentarlo contamos con las siguientes teclas: O, para ir a la derecha. P, para la izquierda. Q, hacia arriba. A, hacia abajo.



```
   1 PAPER 0: BORDER 0: CLS
   2 PRINT INK 6;AT 11,9;"PARA L
A CINTA": GO SUB 9000
   5 GO SUB 9060: GO SUB 8000: G
O SUB 3000
  10 FOR a=1 TO 20 STEP 2: PRINT
 INK 1; PAPER 6;AT 1,a;"■";AT 20
,a;"■": NEXT a
  20 FOR a=2 TO 20 STEP 2: PRINT
 INK 6; PAPER 1;AT 1,a;"■";AT 20
,a;"■": NEXT a
  30 FOR a=1 TO 20 STEP 2: PRINT
 INK 6; PAPER 1;AT a,1;"■";AT a,
20;"■": NEXT a
  40 FOR a=2 TO 20 STEP 2: PRINT
 INK 1; PAPER 6;AT a,1;"■";AT a,
20;"■": NEXT a
  42 PRINT INK 7; FLASH 1;AT 21,
2;"PULSE PARA EMPEZAR": PAUSE 0:
 PRINT AT 21,2;"
                 "

  50 GO SUB 2600


 100 LET vi2=24: LET vi=3: LET t
er=0: LET ter2=49: LET x=7: LET
y=0: LET pl=135: LET a$="AB ": L
ET b$=" CD": LET c$="AB": LET a=
9: LET b=14: LET c=530
 110 IF INKEY$="p" THEN LET c$=b
$: LET c=500
 120 IF INKEY$="o" THEN LET c$=a
$: LET c=510
 130 IF INKEY$="q" THEN LET c=52
0
 140 IF INKEY$="a" THEN LET c=53
0
 142 FOR t=0 TO ve: NEXT t
 145 LET ter=ter+1
 146 IF ter=65 THEN LET ter2=ter
2+1: PLOT 212,ter2: DRAW INK 7;4
,0: LET ter=0
 147 IF ter2=89 THEN GO TO 4000
 148 IF vi=0 THEN GO TO 4010
 149 IF y+x=184 AND b=18 THEN GO
 TO 5000


 160 GO TO c
 300 PRINT INK 4;AT b,a;c$;AT b+
1,a;"  "
 310 IF ATTR (b-1,a)=2 OR ATTR (
b-1,a+2)=2 THEN GO TO 7000
 311 GO TO 330
 320 PRINT INK 4;AT b,a;c$;AT b-
1,a;"  "
 325 IF ATTR (b,a+3)=2 OR ATTR (
b+1,a)=2 THEN GO TO 7000
 330 IF ATTR (b,a)=2 OR ATTR (b,
a+2)=2 THEN GO TO 7000
 340 IF b=19 THEN LET y=y+8: PLO
T 215,pl: DRAW OVER 1;6,0: LET p
l=pl-1: LET b=3: GO SUB 600
 350 IF b=2 THEN LET y=y-8: PLOT
 215,pl: DRAW OVER 1;6,0: LET pl
=pl+1: LET b=18: GO SUB 600
 360 IF a>=18 THEN LET x=x+1: LE
T a=3: GO SUB 600
 370 IF a=1 THEN LET x=x-1: LET
a=17: GO SUB 600


 380 GO TO 110
 500 PRINT INK 4;AT b,a;c$: LET
a=a+1: GO TO 330
 510 PRINT INK 4;AT b,a;c$: LET
a=a-1: GO TO 330
 520 LET b=b-1: GO TO 300
 530 LET b=b+1: GO TO 320
 600 PLOT 216,pl: DRAW INK 7;7,0
: FOR s=2 TO 19: PRINT AT s,2;"
                  ": NEXT s
```



Javier Igual



```
605 RESTORE
610 FOR v=0 TO x+y
620 READ h$
630 NEXT v
635 LET j=VAL h$
640 GO TO j*100+1000
650 DATA "6","10","2","4","0","7","5","16","8","14","0","0","7","13","11","12","6","9","6","2","1","3","5","1","1","6","11","10","9","6","11","9","3","8","9","8","2","8","0","7","6","0","9","6","7","6","0","12","1","4","10","9","1","1","5","1"
660 DATA "8","7","13","2","8","9","13","9","6","11","11","0","10","2","8","7","13","0","7","4","14","0","2","1","13","0","9","5","1","4","10","9","13","6","7","1","3","5","8","7","1","4","9","8","10","14","6","9","8","10","0","2","1","8","0","7"
670 DATA "5","1","4","10","9","5","6","9","1","1","5","1","4","12","8","7","8","9","1","8","7","13","0","12","6","2","13","0","9","1","6","12","8","10","9","4","7","13","12","1","6","11","2","6","11","9","3","1","3","5","6","12","5","4","10","9"
680 DATA "6","12","8","11","11","10","14","7","1","13","0","7","6","9","1","1","15","8","2","8","9","4","9","3"

1000 FOR z=2 TO 5: PRINT INK 2;AT z,2;"[illegible]": NEXT z
1010 PRINT INK 2;AT 6,2;"[illegible]";AT 7,2;"[illegible]";AT 8,2;"[illegible]";AT 9,9;"[illegible]";AT 10,10;"[illegible]";AT 11,11;"EF"
1020 PRINT INK 2;AT 13,2;"[illegible]";AT 13,16;"HG [illegible]";AT 14,2;"[illegible]";AT 14,7;"HG";AT 14,13;"HG [illegible]";AT 15,2;"[illegible]";AT 16,2;"[illegible]"
1030 FOR z=17 TO 19: PRINT INK 2;AT z,2;"[illegible]": NEXT z
1099 RETURN
1100 FOR z=2 TO 19: PRINT INK 2;AT z,2;"[illegible]";AT z,16;"[illegible]": NEXT z
1110 PRINT INK 2;AT 2,5;"[illegible]";AT 2,13;"[illegible]";AT 3,5;"[illegible]";AT 3,14;"[illegible]";AT 4,5;"[illegible]";AT 4,15;"E";AT 5,5;"[illegible]";AT 5,15;"H";AT 6,5;"[illegible]";AT 6,14;"[illegible]";AT 7,5;"[illegible]";AT 7,14;"[illegible]";AT 8,6;"[illegible]";AT 8,15;"E";AT 9,5;"G";AT 10,5;"E";AT 10,15;"[illegible]";AT 11,14;"[illegible]";AT 12,5;"G";AT 12,13;"[illegible]"
1120 PRINT INK 2;AT 13,5;"[illegible]";AT 13,13;"[illegible]";AT 14,5;"[illegible]";AT 14,14;"[illegible]";AT 15,5;"[illegible]";AT 15,13;"[illegible]";AT 16,5;"[illegible]";AT 16,13;"[illegible]";AT 17,5;"[illegible]";AT 17,14;"[illegible]";AT 18,5;"[illegible]";AT 18,14;"[illegible]";AT 19,5;"[illegible]";AT 19,13;"[illegible]"
1199 RETURN
1200 FOR z=2 TO 4: PRINT INK 2;AT z,2;"[illegible]": NEXT z
1210 PRINT INK 2;AT 5,2;"[illegible]";AT 6,2;"[illegible]";AT 6,12;"EF [illegible]";AT 7,2;"[illegible]";AT 7,16;"[illegible]";AT 8,2;"[illegible]";AT 8,15;"[illegible]";AT 9,15;"[illegible]";AT 10,16;"[illegible]";AT 11,17;"[illegible]";AT 12,16;"[illegible]";AT 13,2;"[illegible]";AT 13,16;"[illegible]"
1220 PRINT INK 2;AT 14,2;"[illegible]";AT 14,16;"[illegible]";AT 15,2;"[illegible]";AT 15,15;"[illegible]";AT 16,2;"[illegible]";AT 16,15;"[illegible]";AT 17,2;"[illegible]";AT 18,2;"[illegible]";AT 19,2;"[illegible]"
1299 RETURN
1300 PRINT INK 2;AT 2,2;"[illegible]";AT 2,13;"[illegible]";AT 3,2;"[illegible]";AT 3,16;"[illegible]";AT 4,2;"[illegible]";AT 4,15;"[illegible]";AT 5,2;"[illegible]";AT 5,15;"[illegible]";AT 6,2;"[illegible]";AT 6,16;"[illegible]";AT 7,2;"[illegible]";AT 7,17;"[illegible]";AT 8,2;"[illegible]";AT 8,16;"[illegible]"
1310 PRINT INK 2;AT 9,2;"[illegible]";AT 9,15;"[illegible]";AT 10,2;"[illegible]";AT 10,15;"[illegible]";AT 11,2;"[illegible]";AT 11,16;"[illegible]";AT 12,2;"[illegible]";AT 12,17;"[illegible]";AT 13,2;"[illegible]";AT 13,18;"[illegible]";AT 14,2;"[illegible]";AT 14,14;"HG [illegible]"
1320 PRINT INK 2;AT 15,2;"[illegible]";AT 15,14;"[illegible]";AT 16,2;"[illegible]";AT 16,13;"[illegible]";AT 17,2;"[illegible]";AT 17,9;"HG [illegible]";AT 18,2;"[illegible]";AT 19,2;"[illegible]"
1399 RETURN
1400 FOR z=2 TO 4: PRINT INK 2;AT z,2;"[illegible]": NEXT z
1410 PRINT INK 2;AT 5,2;"[illegible]";AT 6,2;"[illegible]";AT 6,16;"[illegible]";AT 7,2;"[illegible]";AT 7,16;"[illegible]";AT 8,2;"[illegible]";AT 8,16;"[illegible]";AT 10,2;"[illegible]";AT 11,2;"[illegible]";AT 12,2;"[illegible]";AT 13,2;"[illegible]";AT 13,16;"[illegible]";AT 14,2;"[illegible]";AT 14,10;"HG";AT 14,16;"[illegible]"
1420 PRINT INK 2;AT 15,2;"[illegible]";AT 15,9;"[illegible]";AT 16,2;"[illegible]";AT 16,9;"[illegible]"
1430 FOR z=17 TO 19: PRINT INK 2;AT z,2;"[illegible]": NEXT z
1499 RETURN
1500 PRINT INK 2;AT 2,2;"[illegible]";AT 3,2;"[illegible]"
1510 PRINT INK 2;AT 4,2;"[illegible]";AT 4,11;"[illegible]";AT 5,2;"[illegible]";AT 5,12;"EF [illegible]";AT 6,2;"[illegible]";AT 6,16;"[illegible]";AT 7,2;"[illegible]";AT 7,16;"[illegible]";AT 8,2;"[illegible]";AT 8,17;"[illegible]";AT 9,2;"[illegible]";AT 9,18;"[illegible]";AT 10,2;"[illegible]";AT 10,17;"[illegible]";AT 11,2;"[illegible]";AT 11,16;"[illegible]"
1520 PRINT INK 2;AT 12,2;"[illegible]";AT 12,16;"[illegible]";AT 13,2;"[illegible]";AT 13,17;"[illegible]";AT 14,2;"[illegible]";AT 14,17;"[illegible]";AT 15,2;"[illegible]";AT 15,16;"[illegible]";AT 16,2;"[illegible]";AT 16,15;"[illegible]";AT 17,2;"[illegible]";AT 17,14;"[illegible]"
1530 PRINT INK 2;AT 18,2;"[illegible]";AT 18,13;"[illegible]";AT 19,2;"[illegible]";AT 19,13;"[illegible]"
1599 RETURN
1600 FOR z=2 TO 5: PRINT INK 2;AT z,2;"[illegible]": NEXT z
1610 PRINT INK 2;AT 6,2;"[illegible]";AT 7,2;"[illegible]";AT 7,19;"[illegible]";AT 8,2;"[illegible]";AT 8,19;"[illegible]";AT 9,2;"[illegible]";AT 9,9;"EF [illegible]";AT 10,2;"[illegible]";AT 10,11;"EF";AT 11,2;"[illegible]"
1620 PRINT INK 2;AT 12,2;"[illegible]";AT 13,2;"[illegible]";AT 13,18;"[illegible]";AT 14,2;"[illegible]";AT 14,18;"[illegible]";AT 15,2;"[illegible]";AT 15,15;"[illegible]";AT 16,2;"[illegible]";AT 16,15;"[illegible]";AT 17,2;"[illegible]";AT 17,15;"[illegible]";AT 18,2;"[illegible]";AT 18,13;"[illegible]";AT 19,2;"[illegible]";AT 19,13;"[illegible]"
1699 RETURN
1700 FOR z=2 TO 5: PRINT INK 2;AT z,2;"[illegible]": NEXT z
1710 PRINT INK 2;AT 6,2;"[illegible]";AT 7,2;"[illegible]";AT 7,6;"[illegible]";AT 7,17;"[illegible]";AT 8,2;"[illegible]";AT 8,8;"[illegible]";AT 8,17;"[illegible]";AT 9,8;"EF";AT 9,16;"[illegible]"
1720 PRINT INK 2;AT 10,15;"[illegible]";AT 11,15;"[illegible]";AT 12,16;"[illegible]";AT 13,2;"[illegible]";AT 13,17;"[illegible]";AT 14,2;"[illegible]";AT 14,18;"[illegible]";AT 15,2;"[illegible]";AT 15,15;"HG [illegible]";AT 16,2;"[illegible]";AT 16,15;"[illegible]"
1730 PRINT INK 2;AT 17,2;"[illegible]";AT 17,15;"[illegible]";AT 18,2;"[illegible]";AT 18,13;"[illegible]";AT 19,2;"[illegible]";AT 19,13;"[illegible]"
1799 RETURN
1800 PRINT INK 2;AT 2,2;"[illegible]";AT 2,13;"[illegible]";AT 3,2;"[illegible]";AT 3,13;"[illegible]";AT 4,2;"[illegible]";AT 4,13;"[illegible]";AT 5,2;"[illegible]";AT 5,13;"[illegible]";AT 6,2;"[illegible]";AT 6,13;"[illegible]"
1810 PRINT INK 2;AT 7,2;"[illegible]";AT 7,13;"[illegible]";AT 8,2;"[illegible]";AT 8,15;"EF [illegible]";AT 9,2;"[illegible]";AT 10,2;"[illegible]";AT 11,2;"[illegible]";AT 12,2;"[illegible]";AT 13,2;"[illegible]";AT 13,11;"HG";AT 13,19;"[illegible]"
1820 PRINT INK 2;AT 14,2;"[illegible]";AT 14,19;"[illegible]";AT 15,2;"[illegible]";AT 16,2;"[illegible]"
1830 FOR z=17 TO 19: PRINT INK 2;AT z,2;"[illegible]": NEXT z
1899 RETURN
1900 PRINT INK 2;AT 2,2;"[illegible]";AT 2,13;"[illegible]";AT 3,2;"[illegible]";AT 3,13;"[illegible]";AT 4,2;"[illegible]";AT 4,13;"[illegible]";AT 5,2;"[illegible]";AT 5,13;"[illegible]";AT 6,2;"[illegible]";AT 6,13;"[illegible]";AT 7,2;"[illegible]";AT 7,13;"[illegible]"
1910 PRINT INK 2;AT 8,2;"[illegible]";AT 8,15;"[illegible]";AT 9,15;"[illegible]";AT 10,15;"EF [illegible]";AT 11,18;"[illegible]";AT 12,18;"[illegible]";AT 13,2;"[illegible]";AT 13,17;"[illegible]";AT 14,2;"[illegible]";AT 14,16;"[illegible]";AT 15,2;"[illegible]";AT 15,14;"[illegible]";AT 16,2;"[illegible]"
1920 FOR z=17 TO 19: PRINT INK 2;AT z,2;"[illegible]": NEXT z
1999 RETURN

2000 FOR z=2 TO 5: PRINT INK 2;AT z,2;"[illegible]": NEXT z
2010 PRINT INK 2;AT 6,2;"[illegible]";AT 7,2;"[illegible]";AT 8,2;"[illegible] EF";AT 8,12;"[illegible]";AT 9,13;"EF";AT 13,2;"[illegible] HG";AT 13,13;"HG";AT 13,18;"[illegible]"
```

```
2020 PRINT INK 2;AT 14,2;"[redacted]G [redacted]";AT 14,13;"[redacted]GH[redacted]";AT 15,2;"[redacted]";AT 15,13;"[redacted]";AT 16,2;"[redacted]F";AT 16,13;"E[redacted]";AT 17,2;"[redacted]G";AT 17,14;"[redacted]";AT 18,2;"[redacted]G";AT 18,14;"[redacted]";AT 19,2;"[redacted]";AT 19,13;"[redacted]"
2099 RETURN
2100 PRINT INK 2;AT 2,2;"[redacted]";AT 2,13;"[redacted]";AT 3,2;"[redacted]F";AT 3,13;"[redacted]";AT 4,2;"[redacted]F";AT 4,13;"E[redacted]";AT 5,2;"[redacted]";AT 5,13;"H[redacted]";AT 6,2;"[redacted]";AT 6,13;"[redacted]H[redacted]"
2110 PRINT INK 2;AT 7,2;"[redacted]FEF";AT 7,13;"EF[redacted]";AT 8,2;"[redacted]F";AT 8,15;"[redacted]E[redacted]";AT 9,15;"EF";AT 12,11;"HG";AT 13,2;"[redacted]G";AT 13,10;"H[redacted]GHG H[redacted]";AT 14,2;"[redacted]";AT 14,10;"[redacted]";AT 15,2;"[redacted]HG";AT 15,10;"[redacted]";AT 16,2;"[redacted]G H[redacted]"
2120 FOR z=17 TO 19: PRINT INK 2;AT z,2;"[redacted]": NEXT z
2199 RETURN
2200 PRINT INK 2;AT 2,2;"[redacted]";AT 2,13;"[redacted]";AT 3,2;"[redacted]";AT 3,13;"[redacted]";AT 4,2;"[redacted]F";AT 4,13;"E[redacted]";AT 5,2;"[redacted]";AT 5,14;"E[redacted]";AT 6,2;"[redacted]F[redacted]";AT 6,15;"[redacted]";AT 7,2;"[redacted]G EF";AT 7,14;"H[redacted]";AT 8,2;"[redacted]";AT 8,14;"[redacted]"
2210 PRINT INK 2;AT 9,13;"H[redacted]";AT 10,13;"[redacted]";AT 11,13;"EFE[redacted]";AT 12,16;"E[redacted]";AT 13,2;"[redacted]G";AT 13,17;"[redacted]";AT 14,2;"[redacted] HG";AT 14,13;"HG H[redacted]";AT 15,2;"[redacted]HG[redacted]";AT 15,13;"[redacted]"
2220 PRINT INK 2;AT 16,2;"[redacted]";AT 16,13;"E[redacted]";AT 17,2;"[redacted]";AT 17,14;"[redacted]";AT 18,2;"[redacted]";AT 18,13;"H[redacted]";AT 19,2;"[redacted]";AT 19,13;"[redacted]"
2299 RETURN
2300 PRINT INK 2;AT 2,2;"[redacted]";AT 2,13;"[redacted]";AT 3,2;"[redacted]";AT 3,13;"[redacted]";AT 4,2;"[redacted]F";AT 4,13;"[redacted]";AT 5,2;"[redacted]";AT 5,13;"EF[redacted]";AT 6,2;"[redacted]FEF";AT 6,15;"[redacted]"
2310 PRINT INK 2;AT 7,2;"[redacted]";AT 7,15;"EFE[redacted]";AT 8,2;"[redacted]G";AT 8,18;"E[redacted]";AT 9,2;"[redacted]F";AT 10,2;"[redacted]G";AT 11,2;"[redacted]G";AT 12,2;"[redacted]F";AT 12,17;"HG";AT 13,2;"[redacted]F";AT 13,17;"[redacted]";AT 14,2;"[redacted]";AT 14,16;"H[redacted]"
2320 PRINT INK 2;AT 15,2;"[redacted]GHG";AT 15,16;"E[redacted]";AT 16,2;"[redacted]";AT 16,13;"HG [redacted]";AT 17,2;"[redacted]G";AT 17,13;"[redacted]GH[redacted]";AT 18,2;"[redacted]";AT 18,13;"[redacted]";AT 19,2;"[redacted]";AT 19,13;"[redacted]"
2399 RETURN
2400 PRINT INK 2;AT 2,2;"[redacted]";AT 2,13;"[redacted]";AT 3,2;"[redacted]";AT 3,13;"E[redacted]";AT 4,2;"[redacted]F";AT 4,13;"H[redacted]";AT 5,2;"[redacted]FE[redacted]";AT 5,13;"[redacted]F[redacted]";AT 6,2;"[redacted] EF";AT 6,13;"EF[redacted]";AT 7,2;"[redacted]G";AT 7,16;"EFE[redacted]";AT 8,2;"[redacted]";AT 8,19;"[redacted]";AT 9,2;"[redacted]F";AT 9,19;"[redacted]"
2410 PRINT INK 2;AT 13,6;"HG";AT 14,2;"[redacted]GH[redacted]";AT 14,16;"HG [redacted]";AT 15,2;"[redacted]";AT 15,16;"[redacted]H[redacted]";AT 16,2;"[redacted]HG";AT 16,14;"HG[redacted]";AT 17,2;"[redacted]";AT 17,13;"H[redacted]";AT 18,2;"[redacted]";AT 18,13;"[redacted]";AT 19,2;"[redacted]";AT 19,13;"[redacted]"
2499 RETURN
2500 PRINT INK 2;AT 2,2;"[redacted]";AT 2,13;"[redacted]";AT 3,2;"[redacted]FEF";AT 3,13;"EFEFE[redacted]";AT 4,2;"[redacted]F";AT 4,18;"E[redacted]"
2510 FOR z=5 TO 15 STEP 2: PRINT INK 2;AT z,2;"[redacted]G";AT z+1,2;"[redacted]F": NEXT z
2520 FOR z=5 TO 15 STEP 2: PRINT INK 2;AT z,18;"H[redacted]";AT z+1,18;"E[redacted]": NEXT z
2530 PRINT INK 2;AT 17,2;"[redacted]G";AT 17,18;"H[redacted]";AT 18,2;"[redacted]GHGHG";AT 18,13;"HGHGH[redacted]";AT 19,2;"[redacted]"
2540 PRINT INK 2;AT 8,10;"HG";AT 9,9;"H[redacted]G";AT 10,8;"H[redacted]G";AT 11,8;"E[redacted]F";AT 12,9;"E[redacted]F";AT 13,10;"EF"
2545 PRINT INK 3;AT 18,10;"I"
2599 RETURN
2600 PRINT INK 2;AT 8,9;"H[redacted]G";AT 9,8;"HF  EG";AT 10,7;"HF";AT 10,13;"EG";AT 11,6;"H[redacted]G";AT 11,13;"H[redacted]G";AT 12,5;"H[redacted]F";AT 12,13;"E[redacted]G";AT 13,4;"H[redacted]";AT 13,14;"[redacted]G"
2610 PRINT INK 2;AT 14,3;"H[redacted]F";AT 14,14;"E[redacted]G";AT 15,2;"H[redacted]";AT 15,15;"[redacted]G";AT 16,2;"[redacted]HG";AT 16,14;"H[redacted]";AT 17,2;"[redacted]F";AT 17,14;"[redacted]";AT 18,2;"[redacted]G";AT 18,13;"H[redacted]";AT 19,2;"[redacted]";AT 19,13;"[redacted]"
2699 RETURN
3000 PLOT 210,48: DRAW INK 7;0,40: DRAW INK 7;8,1,-PI: PLOT 218,48: DRAW INK 7;0,40
3010 FOR z=40 TO 47: PLOT 208,z: DRAW INK 7;12,0: NEXT z
3020 FOR v=36 TO 39: PLOT 211,v: DRAW INK 7;6,0: NEXT v
3030 PRINT INK 4;AT 19,24;"ABABAB"
3040 RETURN
4010 FOR z=0 TO 20: PRINT #0;"  ": NEXT z: CLS
4030 PRINT INK 4;AT 10,8;"G A M E   O V E R"
4050 PRINT INK 4;AT 21,7;"OTRA PARTIDA? (s/n)"
4060 IF INKEY$="s" THEN GO TO 5
4065 IF INKEY$="n" THEN STOP
4070 GO TO 4060
5000 FOR z=0 TO 19: PRINT INK 5;AT z,0;"[redacted]": NEXT z
5005 FOR z=20 TO 21: PRINT INK 1;AT z,0;"[redacted]": NEXT z
5010 PRINT INK 7; PAPER 5;AT 2,5;"JK";AT 4,21;"JK";AT 3,11;"JK"; INK 0; PAPER 5;AT 18,21;"[redacted]";AT 17,22;"Q";AT 18,26;"E"
5020 PRINT PAPER 5;AT 19,12;"E[redacted]F"; PAPER 1;AT 20,13;"E[redacted]F"
5030 FOR z=26 TO 17 STEP -1: PRINT PAPER 5; INK 1;AT 5,z;"AB ": NEXT z
5040 FOR z=0 TO 13: PAUSE 6: PRINT PAPER 5; INK 5;AT 5+z-1,17;"  ": PRINT PAPER 5; INK 1;AT 5+z,17;"AB ": NEXT z
5100 FOR m=0 TO 12
5200 PRINT PAPER 5;AT 18,21-m;"[redacted] "; INK 1;AT 18,17-m;"AB "; INK 0;AT 17,22-m;"Q ";AT 18,26-m;"E "
5210 PRINT PAPER 5;AT 19,12-m;"E[redacted]F "; PAPER 1;AT 20,13-m;"E[redacted]F "
5220 PAUSE 7: NEXT m
5300 PAUSE 10: BORDER 0: PAPER 0: CLS : GO TO 4050
7000 LET vi=vi-1: PRINT INK 6; BRIGHT 1;AT b,a+1;"LM";AT b+1,a+1;"NO": BEEP .1,00.4: BEEP .01,17: BEEP .01,0: PRINT AT b,a;"  ";AT b+1,a;"  ": LET c=530: LET x=7: LET y=0: GO SUB 600: LET a=9: LET b=14
7001 PLOT 215,pl: DRAW OVER 1;8,0: LET pl=135
7002 PRINT INK 3;AT 19,vi2;"AB": LET vi2=vi2+2: GO TO 110
8000 FOR z=2 TO 7: PRINT INK 6; PAPER 2;AT z,23;"[redacted]";AT z,30;"[redacted]": NEXT z
8010 FOR z=24 TO 29: PRINT INK 6; PAPER 2;AT 1,z;"[redacted]";AT 8,z;"[redacted]": NEXT z
8020 PRINT INK 6; PAPER 2;AT 1,23;"[redacted]";AT 1,30;"[redacted]";AT 8,23;"[redacted]";AT 8,30;"[redacted]"
8030 FOR z=2 TO 7: PRINT INK 5;AT z,24;"[redacted]";AT z,28;"[redacted]": NEXT z
8040 PRINT INK 0; PAPER 5;AT 5,26;"H";AT 5,28;"G";AT 6,25;"H[redacted]";AT 6,28;"[redacted]G";AT 7,24;"H[redacted]";AT 7,28;"[redacted]": PRINT INK 7; PAPER 5;AT 3,25;"JK"
8050 FOR z=2 TO 4: PRINT INK 5;AT z,27;"[redacted]": NEXT z
8060 RETURN
9000 RESTORE 9010: FOR a=144 TO 160: FOR b=0 TO 7: READ c: POKE USR (CHR$ a)+b,c: NEXT b
9010 DATA 0,0,0,3,13,25,127,63,7,15,30,254,95,255,254,252,224,240,120,127,250,255,127,63,0,0,0,192,176,152,254,252
9020 DATA 255,127,63,31,15,7,3,1,255,254,252,248,240,224,192,128,128,192,224,240,248,252,254,255,1,3,7,15,31,63,127,255
9030 DATA 24,24,60,90,90,24,36,36,25,63,111,121,127,239,127,56,240,252,190,94,188,222,232,64
9040 DATA 240,138,68,34,145,73,23,14,82,35,84,202,148,233,135,47,21,85,84,168,178,200,128,236,146,73,58,82,137,71,0,0
9050 DATA 1,3,6,12,60,126,255,255,12,16,32,34,60,32,32,32
9051 NEXT a
9055 RETURN
9060 CLS : PRINT INK 7;AT 8,7;"ELIJA LA VELOCIDAD";AT 13,10;"NORMAL....1";AT 15,10;"RAPIDA....2"
9070 IF INKEY$="1" THEN LET ve=5: CLS : RETURN
9080 IF INKEY$="2" THEN LET ve=0: CLS : RETURN
9090 GO TO 9070
```

Juegos & ESTRATEGIA

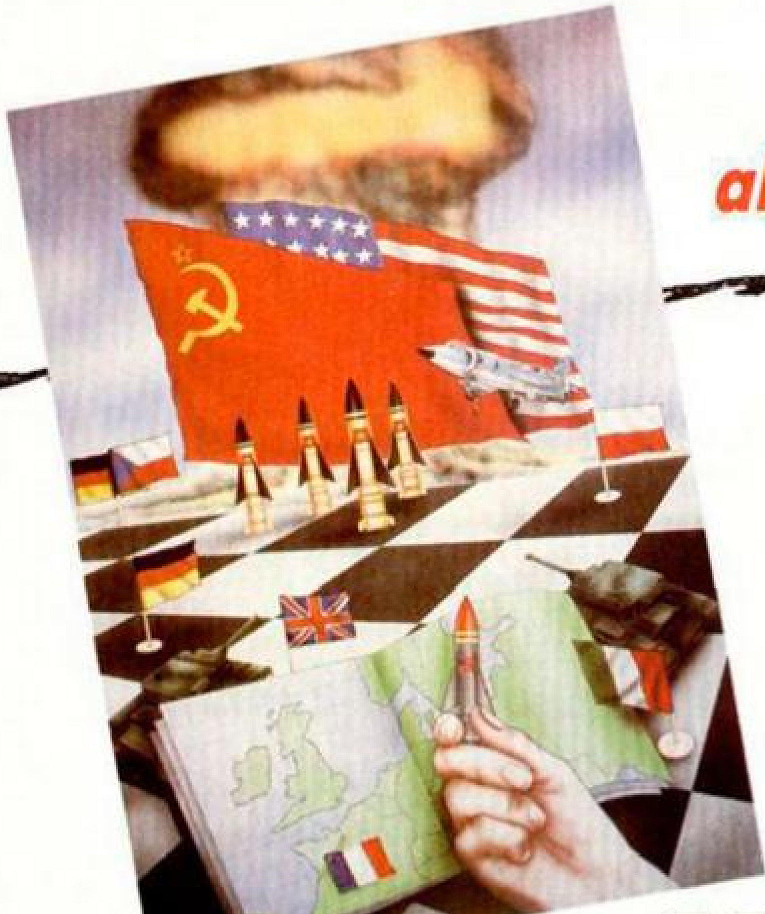
YA ESTA A LA VENTA EL NUMERO 3

Las tropas de la OTAN se enfrentan a las del PACTO DE VARSOVIA

¡TU DIRIGES LAS OPERACIONES!

OFERTA ESPECIAL

3 números al precio de 2



Versión íntegra en castellano.
Incluye pantallas de acción tipo Arcade.



ARHEM
Es Vd. capaz de tomar el relevo del General Montgomery y asumir la estrategia de una batalla que pudo cambiar el curso de la Historia?

RATAS DEL DESIERTO
Se atreve Vd. a dirigir la Campaña del Desierto y derrotar a Rommel antes de lo que lo hizo el General Montgomery en el Alamein?

OTAN ALERTA
Europa es el campo de batalla y tú diriges las operaciones: divisiones de tierra, buques, etc.
¡Debes frenar el avance del Pacto de Varsovia!

Recorta o copia este cupón y envíalo a HOBBY PRESS, S.A. Apdo. 232. Alcobendas (MADRID)

☐ **D**eseo recibir en mi domicilio tres ejemplares de **Juegos & Estrategia al precio especial de 2.255 ptas.** (IVA incluido), lo que me supone adquirir tres y **pagar sólo dos.** (Oferta válida sólo hasta el 31 de mayo de 1986. Marco los tres ejemplares que deseo con una cruz.

Spectrum	Amstrad	Commodore
N.° 1 ☐ Arnhem	☐ Arnhem	
N.° 2 ☐ Ratas del Desierto	☐ Ratas del Desierto	
N.° 3 ☐ OTAN Alerta War Zone	☐ Teatro de Europa War Zone	☐ Teatro de Europa

☐ **D**eseo recibir un solo ejemplar de **Juegos & Estrategia** al precio de 1.125 ptas. (IVA incluido). Marco con una cruz el ejemplar que deseo recibir.

Spectrum	Amstrad	Commodore
N.° 1 ☐ Arnhem	☐ Arnhem	
N.° 2 ☐ Ratas del Desierto	☐ Ratas del Desierto	
N.° 3 ☐ OTAN Alerta War Zone	☐ Teatro de Europa War Zone	☐ Teatro de Europa

NOMBRE ______ EDAD ______
DIRECCION ______
LOCALIDAD ______ PROVINCIA ______
C. POSTAL ______ TELEFONO ______ PROFESION ______

Forma de pago:
☐ Talón bancario a nombre de Hobby Press, S.A.
☐ Giro postal a nombre de Hobby Press, S.A., N.° del giro ______
☐ Tarjeta de crédito: Visa N.° ______ Master Charge N.° ______ American Express N.° ______
Fecha de caducidad de la tarjeta ______ Fecha y firma

Oferta válida
Sólo hasta el 31 de mayo

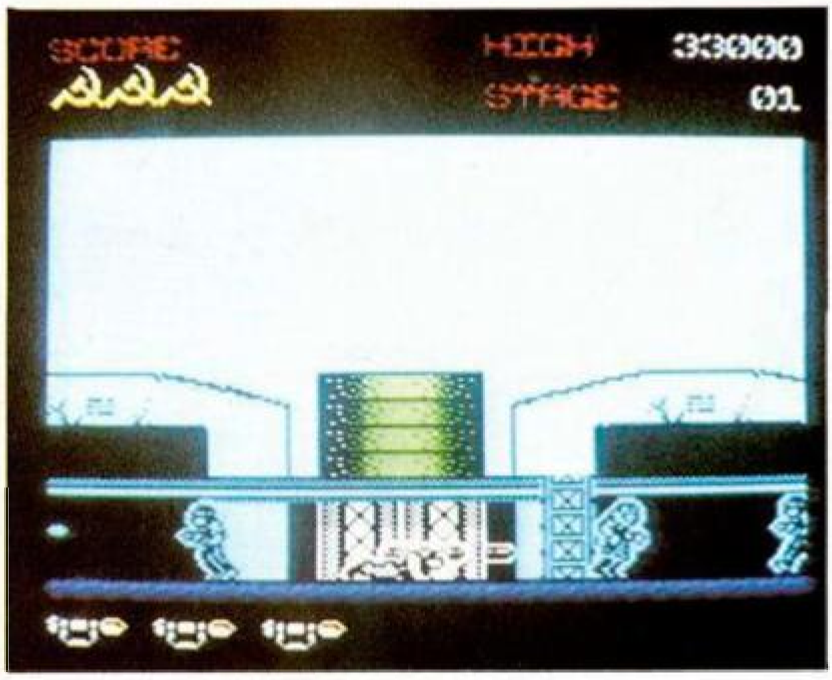
HOBBY PRESS
Para gente inquieta.

GREEN BERET • Arcade • Imagine/Konami

MISION SUICIDA

Imagine continúa realizando versiones para ordenador de los mayores éxitos europeos en juegos recreativos, lo que es una buena forma de asegurarse un éxito de ventas; pero cuando se trata de un juego como Green Beret, sobra cualquier circunstancia que pueda unirse a la elevada calidad que el propio juego aporta.





En esto del software, nunca se acaban de recibir sorpresas. Cuando parece que ya todo está inventado, que ya no queda ningún tema por abordar, que difícilmente se puede encontrar algo que realmente llame la atención, aparece de pronto un programa que incluso basándose en un tema ya bastante tocado y empleando otros recursos que poco o nada tienen de originales, es capaz de despertar el interés de todos.



Estamos hablando, por supuesto, de Green Beret (Boina Verde). Este programa seguramente os resultará familiar a gran parte de vosotros debido a que es uno de los juegos que en la actualidad están obteniendo un mayor éxito en las máquinas de juegos recreativos de toda Europa.

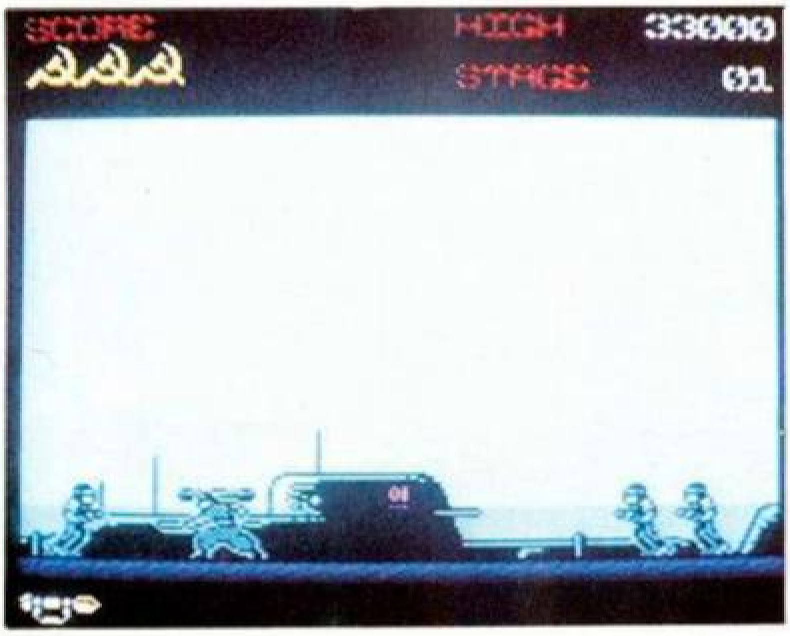
Green Beret, para aquéllos que aún no lo conozcan, pertenece a esa nueva ola de programas que últimamente nos invade cuyo tema principal y casi único se centra en las violentas hazañas bélicas realizadas por un solitario y ultra-heroico soldado, quien se ve envuelto en una misión suicida en la cual debe enfrentarse ante cientos, miles e incluso cientos de miles de soldados que implacables le persiguen con el único objetivo de borrarle del mapa.

Esto, por supuesto que os sonará a programas como Rambo o Comando, aunque existe un gran número de juegos que indirectamente utilizan idénticos alicientes (acción, violencia, tiros, prisioneros, metralletas, lanzallamas, morteros...) ya que, en vista de los resultados obtenidos, parece que son los preferidos por una amplia mayoría de los usuarios. Esto, desde luego, no tiene nada de extraño, ya que efectivamente lo que todo el mundo desea cuando se sienta delante de un ordenador es sentir el mayor número posible de emociones y estos programas, no cabe la menor duda de que las ofrecen.

Pero, ¿qué es lo que hace que Green Beret destaque con respecto al resto de sus enérgicos competidores?

Pues principalmente dos únicas razones. La primera de ellas es la alta calidad de sus gráficos. Estos están confeccionados con un gran lujo de detalles y tanto los personajes como los distintos escenarios por los que nos vamos moviendo resultan bastante atractivos y originales.

Otra característica muy peculiar con respecto a los aspectos gráficos es que, a

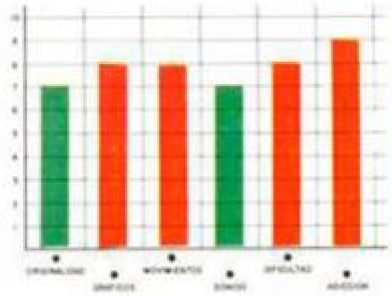


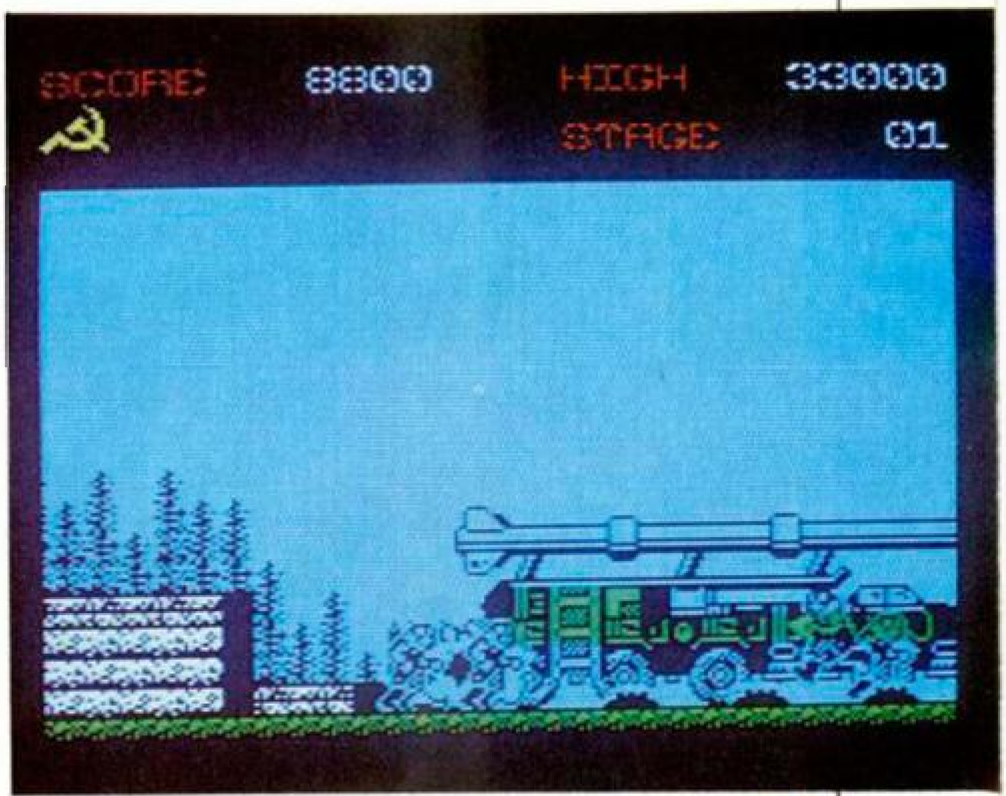


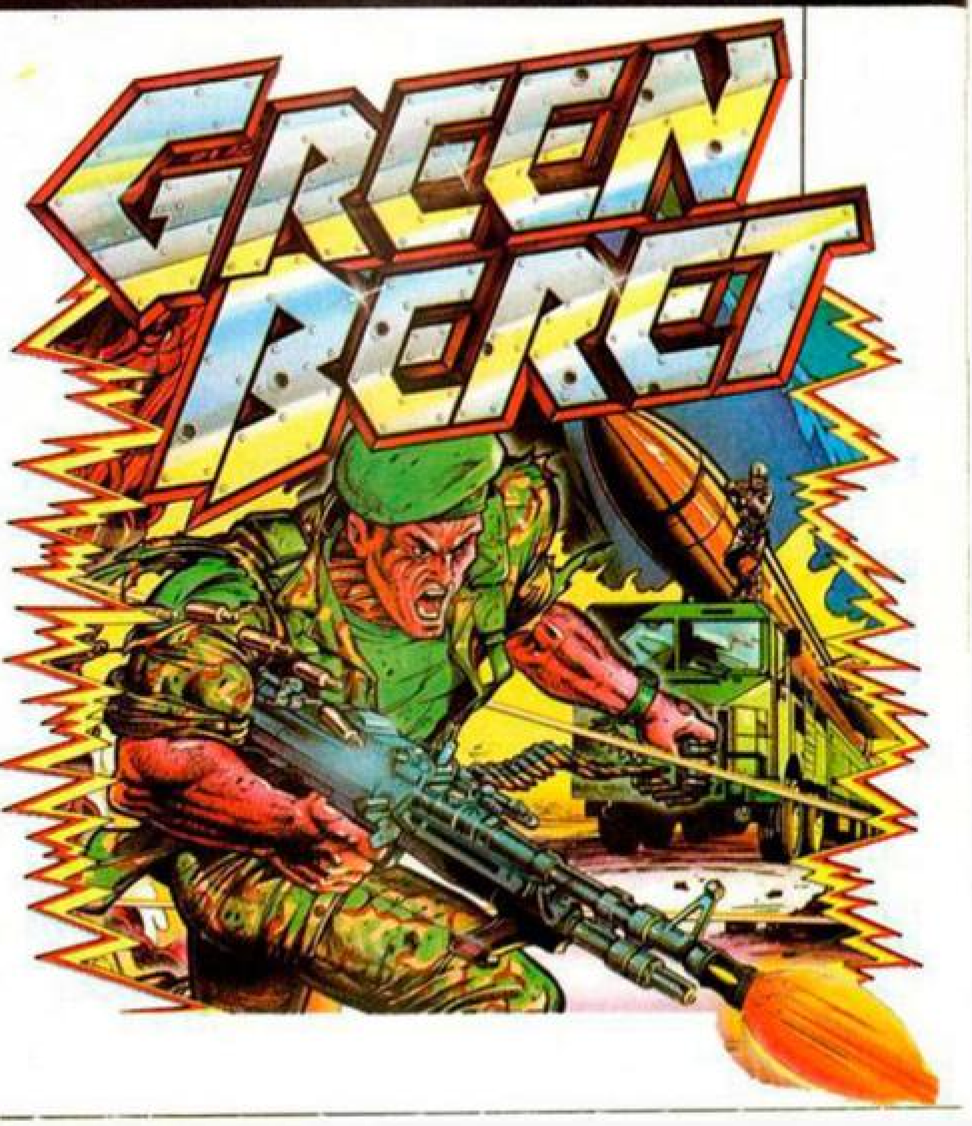
diferencia de lo que ocurre en Comando o Rambo, la acción transcurre en sentido horizontal con respecto a la pantalla, lo que hace que este valeroso Boina Verde pueda moverse hacia arriba o hacia abajo para intentar esquivar a sus atacantes.

La otra circunstancia que hace que destaque, (más que a un nivel de calidad nos referimos al nivel de adicción), es que a pesar de que este juego es de una acción bastante intensa, no llega a ser tan agobiante como la de otros programas similares. Lo que queremos decir es que, debido a la disposición horizontal de los personajes tan sólo podemos ser atacados bien por la izquierda o bien por la derecha, con lo que tendremos (más o menos), el tiempo suficiente para reaccionar y dirigir nuestro ataque en la dirección que sea necesario. De este modo se logra que, a pesar de que los ataques son incesantes y de muy diferente naturaleza (disparos, patadas, cañonazos, etc...), podemos pensar, aunque sea en una milésima de segundo, cuál es la acción defensiva que vamos a realizar.

Y esto es prácticamente todo cuanto podemos contaros a cerca de Green Beret, pues como podréis suponer, se trata de un juego en el que prácticamente no es demasiado necesario el plantearse ninguna estrategia especial, ni darle vueltas a la cabeza buscando la utilidad de ningún objeto, ni tampoco ver si vamos a poder pasar por esa o aquella puerta. Aquí tan sólo tendremos que limitarnos a movernos a toda velocidad y con gran astucia hasta lograr nuestro objetivo final: divertirnos en cantidad.







VECTRON • Arcade • Firebird

EL LABERINTO DEL ROBOT

Los laberintos se están convirtiendo en toda una institución en esto de los programas para ordenador. Desde luego, de lo que no cabe la menor duda es de que existe un número muy elevado de juegos en el que se utilizan estos complicados y enmarañados caminos como parte fundamental en el desarrollo de los mismos.

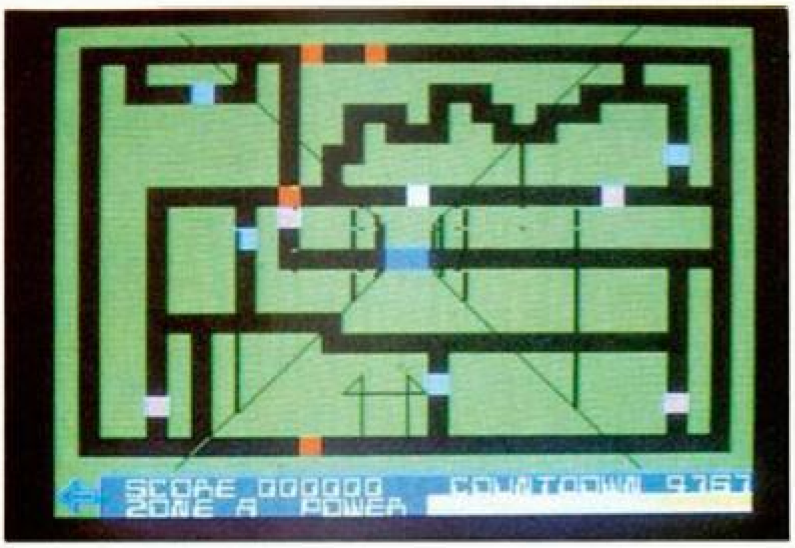
Y la verdad es que los tenemos para todos los gustos: formados por espesa y verde vegatación, como en Robin of the wood; por ricos y enormes setos como en Saimazoon; por largas y tortuosas calles como en Dun Darach; por oscuras grutas como en Starquake, etc... Sin embargo, este Vectron está realizado en la línea del puro y liso laberinto, es decir, en el que no hay nada especial que forme los límites del mismo sino que está confeccionado por simples rayas que definen las paredes. Un claro ejemplo de esto podría ser el conocido «I, of the mask», aunque quizá Vectron resulte de una calidad gráfica sensiblemente inferior.

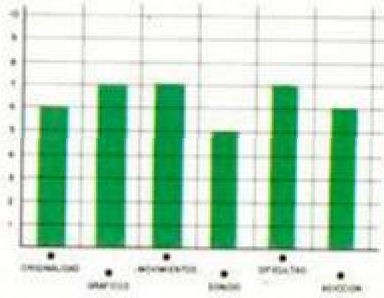
El objetivo del juego consiste en llegar hasta el centro del laberinto y encontrar y destruir el Rom Robot. Después deberemos preparar la escapada, pero nuestras condiciones serán mucho peores que en los primeros momentos, ya que no dispondremos del mapa con el que guiarnos en la primera fase y además, nuestro estado físico será bastante más flojo, puesto que habremos perdido gran parte de nuestras fuerzas en este largo camino.

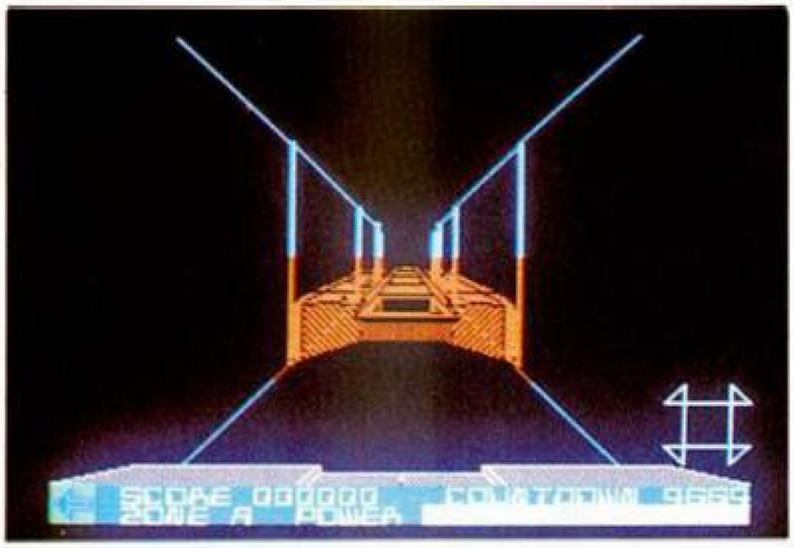
La dificultad del juego estriba principalmente en que esta energía no se nos agote por completo, ya que al mínimo choque con cualquiera de las paredes perderemos una buena parte de ella. Estos choques, sin embargo, serán prácticamente inevitables, ya que debido a la gran velocidad a la que nos movemos por el laberinto apenas tendremos tiempo para reaccionar.

Y por si acaso nos parece monótono el viajecito, también irán saliendo a nuestro encuentro algunas naves enemigas, contra las cuales, por supuesto, tendremos que disparar.

En cuanto a la valoración estética de este Vectron, simplemente señalar que la sensación de movimiento está muy bien conseguida, pero los escenarios son un tanto pobres, aunque de buena realización.







«V» • Arcade • Ocean

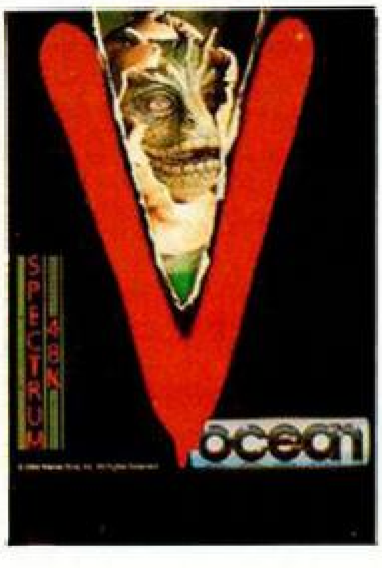
ACABA CON LA MALDICION DE LOS LAGARTOS

Hace ya algunos meses que dejó de emitirse por televisión aquella popular serie en la cual unos extraños visitantes, mitad humanos mitad reptiles, intentaban invadir y dominar el planeta Tierra.

Seguro que ya habéis reconocido de qué serie estamos hablando. Efectivamente, se trata de «V». Pues bien, no creáis que ahora nos hemos convertido en una revista de programación televisiva, sino que la peliculilla en cuestión vuelve a estar de moda porque Ocean ha tenido a bien realizar un juego basado en el argumento y personajes de dicha serie.

El objetivo del juego consiste, por supuesto, en acabar definitivamente con la amenaza de los visitantes para lo cual tendremos que, representando el papel del heroico Donovan, hacer explotar cuatro explosivos en diferentes instalaciones de la enorme y complicada nave nodriza de los visitantes, en el interior de la cual nos encontramos.

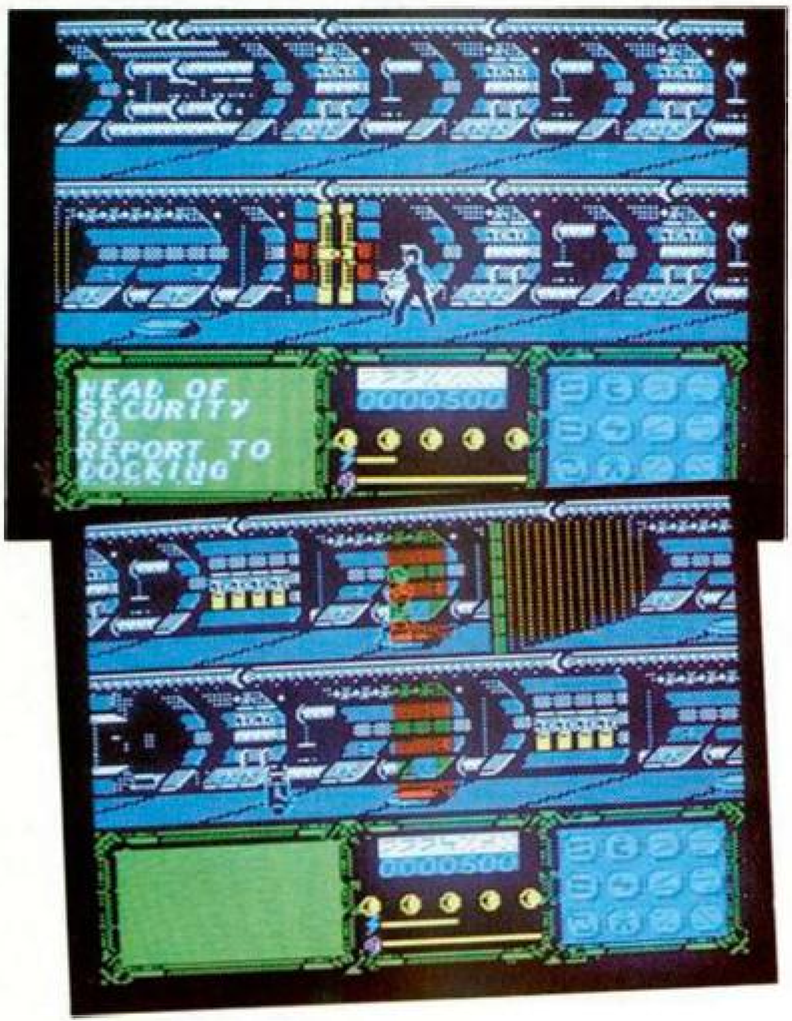
Pero para complicar más las cosas, parece ser que la malvada Diana, una de las lagartas más guapas que se han visto por aquí, ya se ha enterado de nuestra presen-

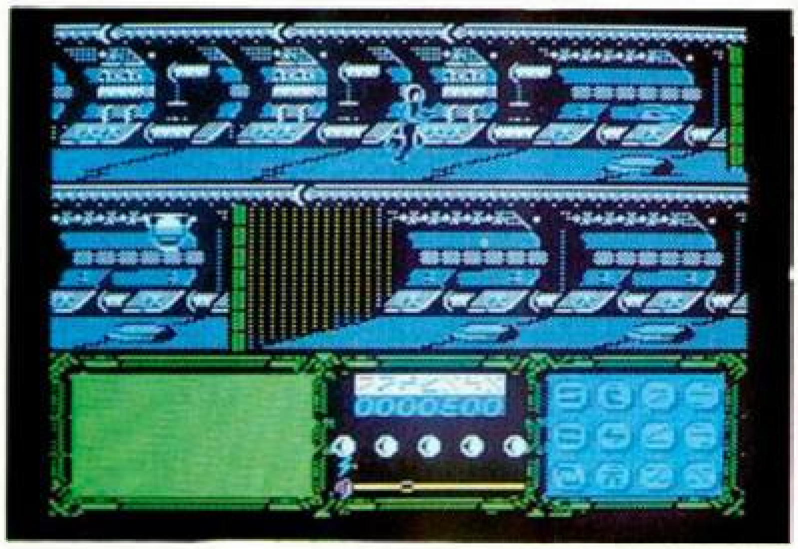


cia, por lo que ha dado la señal de alarma y ha puesto a un nutrido ejército de robots siguiéndonos los pasos. Estos guardianes metálicos son de varios tipos diferentes y aunque ninguno de ellos nos eliminará instantáneamente, sí que irán haciendo que poco a poco nuestro corazón vaya latiendo a más velocidad, lo que hará que cada vez nos vayamos moviendo con más lentitud hasta que nos llegue una muerte dulce y tranquila.

Sin embargo, no estaremos indefensos ante los ataques de estos artefactos mecánicos, pues algunos de ellos serán buen blanco de prácticas para demostrar nuestra puntería y nuestras habilidades.

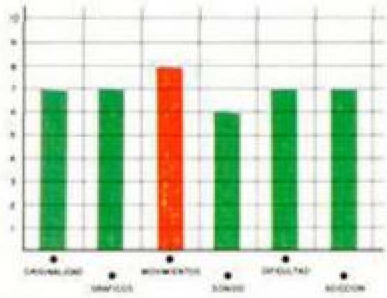
Por otra parte, también podremos realizar otros tipos de movimientos y acciones acudiendo a un pequeño grupo de iconos que aparecen en la parte inferior de la pantalla. Con ellos podremos activar los explosivos, elegir las claves de seguridad, etc... Lo que ocurre es que todas estas acciones vienen señaladas en un extraño lenguaje intergaláctico que tardaremos algún tiempo en llegar a comprender, pero que una vez que así lo hagamos, nos será de una vital importancia.





La nave es prácticamente igual en todas sus habitaciones y pasadizos, por lo que quizá resulte un tanto monótono gráficamente, sin embargo, esto se contrarresta con el placer de ver correr y saltar al atlético Donovan, cuyos movimientos están realizados con suma perfección, recordando bastante a los del protagonista de Impossible Mission.

Un buen juego, mezcla de acción y algo de estrategia con el que pasaremos buenos ratos intentando destruir a estos feos y viscosos visitantes que ya nos entretuvieron muchas veces en la pequeña pantalla.



BLADE RUNNER • Arcade • CRL

LOS ANDROIDES INVADEN LA TIERRA

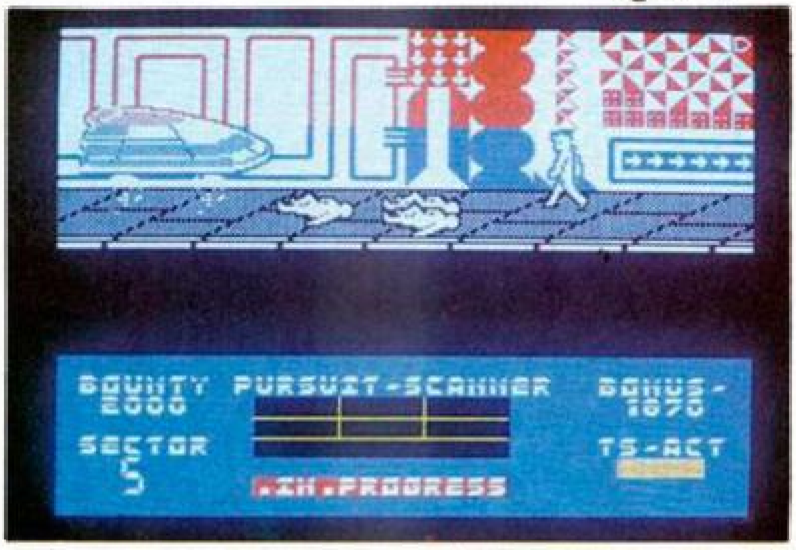
Y como parece que la cosa va de películas esta semana, ahí va el lanzamiento de un nuevo juego basado en el argumento de una gran obra cinematográfica: Blade Runner.

Ambas comparten el mismo tema: Los Angeles. Año 2019. Dick Deckard (ex policía, ex blade runner, ex asesino), acaba de ser reclutado para terminar con cuatro reproducciones Nexus-6, perfectas, de carne y hueso, que han vuelto a la Tierra en busca de supervivencia. Estos son: Roy Batty, inteligente y eficaz, posiblemente el jefe; León Kowalski, una máquina de guerra, sin cerebro; Pris, un modelo básico de placer, morirá pronto si no consigue el secreto para prolongar su vida; y por último, Zhora, la bella y la bestia unidas en una sola pieza.



Estos cuatro personajes han llegado a la Tierra. Son

¡NUEVO!



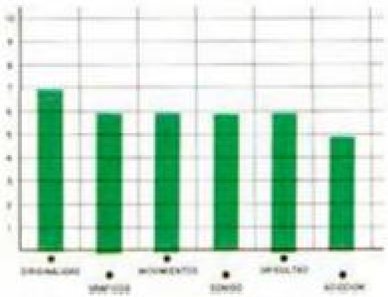
perfectos, tienen inteligencia y en la superficie de nuestro planeta triplican su fuerza. Son muy peligrosos. Saben que sólo les queda vida para cuatro años y están dispuestos a conseguir a toda costa la inmortalidad.

Deckar deberá buscarlos por toda la ciudad y acabar con cada uno de ellos. Pero la misión es difícil y cuenta con muy pocas ayudas. Tan sólo con un pequeño detector de Nexus superespecializado que le permitirá localizar la situación exacta de cada uno de estos androides, y de un rápido y potente Spinner (sofisticado vehículo volador) con el que podrá sobrevolar entre los edificios en busca de sus perseguidos.

Sin embargo, nada de esto podrá servirle de mucha utilidad en los momentos clave en que, por fin, haya de enfrentarse cara a cara con los Nexus. Aquí tan sólo dispondrá de su rapidez y astucia para esquivar los obstáculos que se interpongan en su frenética carrera.

Interesante, ¿verdad? Pues esto es en lo que consiste este Blade Runner. Como veis el guión es bastante atractivo, pero en realidad el resultado final del juego no lo es tanto. Y no por sus aspectos gráficos, los cuales son de una calidad aceptable, sino porque el desarrollo es bastante monótono y, de las dos partes en las que se divide fundamentalmente cada una de las persecuciones, sólo hay una que llama un poco la atención y que resulta algo entretenida, pero tampoco en exceso.

En resumen, un juego que intenta ampararse en el éxito de esta maravillosa película, pero que no ha conseguido aproximarse ni por lo más remoto a la gran calidad de ésta.





FORBIDDEN PLANET • Arcade • Design Design

EL MENSAJE DE LOS SEÑORES DEL MAL

Aparentemente el desarrollo de este juego debe ser bastante sencillo, pues en las instrucciones tan sólo se nos indica que tenemos que localizar las partes del mensaje final de los Señores del Mal, las cuales se encuentran esparcidas por la superficie de este Forbidden Planet, lugar donde nos encontramos sentados a los mandos de nuestro avanzado y sofisticado tanque de batalla.

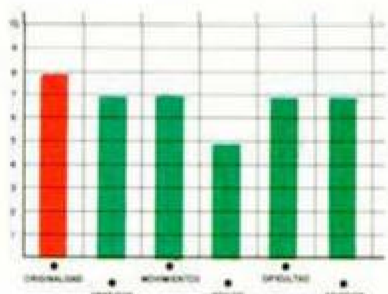


Pero la verdad es que nosotros no lo encontramos sencillo ni muchísimo menos. El descubrir qué es lo que hay que hacer ya cuesta bastante, pero encima es que apenas te queda tiempo para hacer nada, pues después de las primeras escaramuzas, es casi imposible que no te maten con esos rayos que surgen, imparables, de sus torres de defensa.



Sin embargo, no sabemos qué es lo que tiene este Forbidden Planet que, a pesar de esto y de que gráficamente tampoco es ninguna maravilla, resulta un juego de lo más gracioso, y sobre todo cuando pulsamos a fondo el acelerador y nos movemos a unas velocidades alucinógenas. Luego, apretando ciertas teclas, no sabemos cómo ni cuándo, apareces en unos lugares muy extraños donde te persiguen unos signos matemáticos mientras tú te mueves a una velocidad vertiginosa por la pantalla intentando recoger combustible.

En fin, que sobre el argumento del juego poco os podemos decir porque lo interesante es ir descubriéndolo; pero sí os podemos asegurar que resulta de lo más movido, entretenido y gracioso el jugar con este Forbidden Planet, aunque sinceramente creemos que el llegar a completar los códigos perdidos, puede resultar una misión poco menos que imposible. Pero bueno, tampoco importa demasiado si mientras lo intentamos nos lo pasamos bien, ¿no?, que es de lo que se trata.



mente, no existe un algoritmo de cálculo que sea válido para toda la pantalla (al menos, nosotros no lo hemos encontrado, pero si algún lector lo encuentra, no deje de comunicárnoslo).

La solución que nosotros proponemos para este problema, no tiene por qué ser, necesariamente, la única ni siquiera la mejor (en informática, como en casi todos los aspectos de la vida, no existen verdades absolutas), pero al menos, funciona de forma bastante aceptable y permite adaptarla para otros usos.

De momento, imaginemos que numeramos los scans de «0» a «191»; el primer scan de la pantalla, empezando por arriba, sería el scan «0» y el último sería el scan «191». Ahora, el problema se reduce a escribir dos rutinas; una que nos sintetice el número de scan a partir de una dirección dada y la otra, que nos sintetice una dirección a partir del número de scan. Por supuesto, esta segunda rutina no deberá afectar a los bits de la dirección que definen el número de columna, para poder cumplir el requisito de que funcione con un octeto perteneciente a cualquier columna de la pantalla.

Una vez que consigamos tener el número de scan, podemos incrementarlo para pasar al scan siguiente o decrementarlo para pasar al anterior. Aún podemos sacar más partido de este procedimiento: si, al decrementar el número de scan, éste pasase de valer «0» a valer «255» o si, al incrementarlo, pasase de valer «191» a valer «192», la nueva dirección obtenida caería fuera del archivo de pantalla, por lo que sería errónea y nos indicaría que la dirección de partida pertenecía al primer o último scan respectivamente.

Recordemos, del capítulo anterior, el formato de una dirección de pantalla:

```
   BYTE MAS SIGNIFICATIVO
0  1  0  L4 L3 S2 S1 S0
FIJO     LINEA    SCAN

   BYTE MENOS SIGNIFICATIVO
L2 L1 L0 C4 C3 C2 C1 C0
LINEA       COLUMNA
```

Para obtener el número de scan a partir de aquí, no tendremos más que colocar los bits que definen la línea y el scan, en el orden adecuado:

```
      NUMERO DE SCAN
L4 L3 L2 L1 L0 S2 S1 S0
    LINEA        SCAN
```

Supongamos que tenemos la dirección de pantalla en el registro «HL» y queremos formar el número de scan en el registro «A». Transportaremos los bits como se muestra en la Figura 10-7. La rutina para hacerlo, podría ser la siguiente:

```
140 SIG_1  LD   A,H
150        AND  #07
160        LD   B,A
170        LD   A,H
180        AND  #18
190        SLA  A
200        SLA  A
210        SLA  A
220        OR   B
230        LD   B,A
240        LD   A,L
250        AND  #E0
260        SRL  A
270        SRL  A
280        OR   B
```

Operamos con el registro «A» y utilizamos el «B» como almacenamiento temporal. Vamos a ir viendo el funcionamiento línea a línea:

140: Cargamos, en «A», el byte más significativo de la dirección.

150: Seleccionamos, sólo, los tres bits de la derecha, es decir, los que nos definen el número de scan dentro de la línea. Los restantes bits quedan a «0».

160: Transferimos el resultado al registro «B».

170: De nuevo, cargamos, en «A», el byte alto de la dirección.

180: Esta vez aislamos los bits 3 y 4 que definen a qué tercio de pantalla corresponde la dirección.

190: En esta línea y las dos siguientes rotamos estos dos bits a la izquierda tres posiciones.

220: Los mezclamos con los tres que habíamos almacenado anteriormente en «B».

230: Y volvemos a transferir el resultado a «B».

240: Ahora, cargamos en «A» el byte inferior de la dirección.

250: Aislamos los tres bits de más peso que nos indican el número de línea dentro de un tercio determinado.

260: En esta línea y la siguiente, los rotamos dos veces a la derecha para que encajen en los

INSTRUCCIONES DE PUESTA A "1" (I)

Código Fuente	Hexadecimal	Decimal
SET 0,A	CB,C7	203,199
SET 0,B	CB,C0	203,192
SET 0,C	CB,C1	203,193
SET 0,D	CB,C2	203,194
SET 0,E	CB,C3	203,195
SET 0,H	CB,C4	203,196
SET 0,L	CB,C5	203,197
SET 0,(HL)	CB,C6	203,198
SET 0,(IX+d)	DD,CB,d,C6	221,203,d,198
SET 0,(IY+d)	FD,CB,d,C6	253,203,d,198
SET 1,A	CB,CF	203,207
SET 1,B	CB,C8	203,200
SET 1,C	CB,C9	203,201
SET 1,D	CB,CA	203,202
SET 1,E	CB,CB	203,203
SET 1,H	CB,CC	203,204
SET 1,L	CB,CD	203,205
SET 1,(HL)	CB,CE	203,206
SET 1,(IX+d)	DD,CB,d,CE	221,203,d,206
SET 1,(IY+d)	FD,CB,d,CE	253,203,d,206
SET 2,A	CB,D7	203,215
SET 2,B	CB,D0	203,208
SET 2,C	CB,D1	203,209
SET 2,D	CB,D2	203,210
SET 2,E	CB,D3	203,211
SET 2,H	CB,D4	203,212
SET 2,L	CB,D5	203,213
SET 2,(HL)	CB,D6	203,214
SET 2,(IX+d)	DD,CB,d,D6	221,203,d,214
SET 2,(IY+d)	FD,CB,d,D6	253,203,d,214
SET 3,A	CB,DF	203,223
SET 3,B	CB,D8	203,216
SET 3,C	CB,D9	203,217
SET 3,D	CB,DA	203,218
SET 3,E	CB,DB	203,219
SET 3,H	CB,DC	203,220
SET 3,L	CB,DD	203,221
SET 3,(HL)	CB,DE	203,222
SET 3,(IX+d)	DD,CB,d,DE	221,203,d,222
SET 3,(IY+d)	FD,CB,d,DE	253,203,d,222

INSTRUCCIONES DE PUESTA A "1" (II)

Código Fuente	Hexadecimal	Decimal
SET 4,A	CB,E7	203,231
SET 4,B	CB,E0	203,224
SET 4,C	CB,E1	203,225
SET 4,D	CB,E2	203,226
SET 4,E	CB,E3	203,227
SET 4,H	CB,E4	203,228
SET 4,L	CB,E5	203,229
SET 4,(HL)	CB,E6	203,230
SET 4,(IX+d)	DD,CB,d,E6	221,203,d,230
SET 4,(IY+d)	FD,CB,d,E6	253,203,d,230
SET 5,A	CB,EF	203,239
SET 5,B	CB,E8	203,232
SET 5,C	CB,E9	203,233
SET 5,D	CB,EA	203,234
SET 5,E	CB,EB	203,235
SET 5,H	CB,EC	203,236
SET 5,L	CB,ED	203,237
SET 5,(HL)	CB,EE	203,238
SET 5,(IX+d)	DD,CB,d,EE	221,203,d,238
SET 5,(IY+d)	FD,CB,d,EE	253,203,d,238
SET 6,A	CB,F7	203,247
SET 6,B	CB,F0	203,240
SET 6,C	CB,F1	203,241
SET 6,D	CB,F2	203,242
SET 6,E	CB,F3	203,243
SET 6,H	CB,F4	203,244
SET 6,L	CB,F5	203,245
SET 6,(HL)	CB,F6	203,246
SET 6,(IX+d)	DD,CB,d,F6	221,203,d,246
SET 6,(IY+d)	FD,CB,d,F6	253,203,d,246
SET 7,A	CB,FF	203,255
SET 7,B	CB,F8	203,248
SET 7,C	CB,F9	203,249
SET 7,D	CB,FA	203,250
SET 7,E	CB,FB	203,251
SET 7,H	CB,FC	203,252
SET 7,L	CB,FD	203,253
SET 7,(HL)	CB,FE	203,254
SET 7,(IX+d)	DD,CB,d,FE	221,203,d,254
SET 7,(IY+d)	FD,CB,d,FE	253,203,d,254

Fig. 10-2. Tabla de codificación para las instrucciones de puesta a «1» de bits.

INSTRUCCIONES DE PUESTA A "0" (I)

Código Fuente	Hexadecimal	Decimal
RES 0,A	CB,87	203,135
RES 0,B	CB,80	203,128
RES 0,C	CB,81	203,129
RES 0,D	CB,82	203,130
RES 0,E	CB,83	203,131
RES 0,H	CB,84	203,132
RES 0,L	CB,85	203,133
RES 0,(HL)	CB,86	203,134
RES 0,(IX+d)	DD,CB,d,86	221,203,d,134
RES 0,(IY+d)	FD,CB,d,86	253,203,d,134
RES 1,A	CB,8F	203,143
RES 1,B	CB,88	203,136
RES 1,C	CB,89	203,137
RES 1,D	CB,8A	203,138
RES 1,E	CB,8B	203,139
RES 1,H	CB,8C	203,140
RES 1,L	CB,8D	203,141
RES 1,(HL)	CB,8E	203,142
RES 1,(IX+d)	DD,CB,d,8E	221,203,d,142
RES 1,(IY+d)	FD,CB,d,8E	253,203,d,142
RES 2,A	CB,97	203,151
RES 2,B	CB,90	203,144
RES 2,C	CB,91	203,145
RES 2,D	CB,92	203,146
RES 2,E	CB,93	203,147
RES 2,H	CB,94	203,148
RES 2,L	CB,95	203,149
RES 2,(HL)	CB,96	203,150
RES 2,(IX+d)	DD,CB,d,96	221,203,d,150
RES 2,(IY+d)	FD,CB,d,96	253,203,d,150
RES 3,A	CB,9F	203,159
RES 3,B	CB,98	203,152
RES 3,C	CB,99	203,153
RES 3,D	CB,9A	203,154
RES 3,E	CB,9B	203,155
RES 3,H	CB,9C	203,156
RES 3,L	CB,9D	203,157
RES 3,(HL)	CB,9E	203,158
RES 3,(IX+d)	DD,CB,d,9E	221,203,d,158
RES 3,(IY+d)	FD,CB,d,9E	253,203,d,158

INSTRUCCIONES DE PUESTA A "0" (II)

Código Fuente	Hexadecimal	Decimal
RES 4,A	CB,A7	203,167
RES 4,B	CB,A0	203,160
RES 4,C	CB,A1	203,161
RES 4,D	CB,A2	203,162
RES 4,E	CB,A3	203,163
RES 4,H	CB,A4	203,164
RES 4,L	CB,A5	203,165
RES 4,(HL)	CB,A6	203,166
RES 4,(IX+d)	DD,CB,d,A6	221,203,d,166
RES 4,(IY+d)	FD,CB,d,A6	253,203,d,166
RES 5,A	CB,AF	203,175
RES 5,B	CB,A8	203,168
RES 5,C	CB,A9	203,169
RES 5,D	CB,AA	203,170
RES 5,E	CB,AB	203,171
RES 5,H	CB,AC	203,172
RES 5,L	CB,AD	203,173
RES 5,(HL)	CB,AE	203,174
RES 5,(IX+d)	DD,CB,d,AE	221,203,d,174
RES 5,(IY+d)	FD,CB,d,AE	253,203,d,174
RES 6,A	CB,B7	203,183
RES 6,B	CB,B0	203,176
RES 6,C	CB,B1	203,177
RES 6,D	CB,B2	203,178
RES 6,E	CB,B3	203,179
RES 6,H	CB,B4	203,180
RES 6,L	CB,B5	203,181
RES 6,(HL)	CB,B6	203,182
RES 6,(IX+d)	DD,CB,d,B6	221,203,d,182
RES 6,(IY+d)	FD,CB,d,B6	253,203,d,182
RES 7,A	CB,BF	203,191
RES 7,B	CB,B8	203,184
RES 7,C	CB,B9	203,185
RES 7,D	CB,BA	203,186
RES 7,E	CB,BB	203,187
RES 7,H	CB,BC	203,188
RES 7,L	CB,BD	203,189
RES 7,(HL)	CB,BE	203,190
RES 7,(IX+d)	DD,CB,d,BE	221,203,d,190
RES 7,(IY+d)	FD,CB,d,BE	253,203,d,190

Fig. 10-3. Tabla de codificación para las instrucciones de puesta a «0» de bits.

fuera del bit 0 del registro «E»; podríamos añadir, entre la línea 300 y la 310, las siguientes instrucciones:

```
304      BIT  0,E
306      JR   Z,NOCA_2
```

Antes de entrar en la rutina, deberíamos fijar el flag a «0» o «1» en función de qué tipo de scroll necesitemos; «SET 1,E» si queremos «scroll esférico» o «RES 1,E» si queremos «scroll lineal».

Una vez visto el scroll a derecha, para hacer la rutina que lo realice a izquierda no tenemos más que copiar ésta cambiando algunas instrucciones:

```
70         LD   HL,22527
80         LD   C,192
90  BUC_2  LD   B,32
100        AND  A
110 BUC_1  RL   (HL)
120        DEC  HL
130        DJNZ BUC_1
140        JR   NC,NOCA_1
150        LD   (VAR),HL
160        LD   IX,(VAR)
170        SET  0,(IX+32)
180 NOCA_1 DEC  C
190        JR   NZ,BUC_2
200        RET
```

En la línea 70 inicializamos el puntero a la última dirección de pantalla, ya que este scroll lo vamos a hacer de abajo a arriba. En la 110 rotamos a la izquierda en lugar de hacerlo a la derecha. En 120 decrementamos en vez de incrementar igual que en 180. Finalmente, en la línea 170, el bit que ponemos a «1» es el último del scan en lugar de el primero. Por lo demás, la rutina es igual que la anterior y sólo se diferencia en las etiquetas.

De nuevo, si no queremos «scroll esférico», podemos eliminar las líneas 140, 150, 160 y 170 o insertar las siguientes:

```
134      BIT  0,E
136      JR   Z,NOCA_1
```

Para controlarlo mediante el mismo flag que en la rutina anterior. Aunque nada impide utilizar flags distintos para cada rutina con lo que podríamos hacer girar la pantalla de forma «esférica» en un sentido y de forma «lineal» en otro.

Como se ve, las posibilidades de utilización de estas rutinas sólo están limitadas por la imaginación del usuario. En cualquier caso, es importante escribir las rutinas que han de formar parte de una biblioteca, de forma que sean fácilmente modificables para adaptarlas a cada uso concreto que se les quiera dar.

Tal vez a algún lector le interese saber cómo puede adaptar estas rutinas para que realicen el «scroll» de sólo un tercio de la pantalla. La adaptación no es difícil, y hemos preferido incluirla como uno de los ejercicios de este capítulo. Le recomendamos, por tanto, que intente resolverlo por sí mismo antes de mirar la solución.

El ensamblado de estas rutinas no presenta ningún problema. Le recomendamos que lo haga en hexadecimal y lo coloque en el formato del «CARGADOR UNIVERSAL» (10 octetos por línea con suma de comprobación al final). Si lo hace así, puede comprobar el resultado en las líneas 310 a 360 del programa Basic que las maneja y que publicamos en este capítulo con el título de «SCROLL TOTAL».

Vamos ahora con el scroll vertical. El planteamiento es el mismo de antes: tenemos que subir la pantalla un scan hacia arriba de forma que el primer scan pase a ser el último, el segundo pase al primero, el tercero al segundo y así sucesivamente. El sistema consiste en coger el primer scan, almacenarlo en otro lugar de la memoria (por ejemplo, el buffer de impresora), entrar en un bucle que, a partir del segundo scan, transfiera cada uno al anterior y, finalmente, deberemos recuperar el scan que guardamos en otra zona de memoria y transferirlo al lugar del último (esto, sólo si queremos que el scroll se realice de forma «esférica»). La solución sería muy sencilla si las direcciones de scans consecutivos fueran también consecutivas; por desgracia, esto no es así.

Lo primero que necesitamos es una rutina que nos dé la dirección del scan siguiente a uno dado. Para poder utilizar esta rutina en otros programas, podemos hacerla de forma que no nos de siempre la primera dirección del scan, sino la del octeto que se encuentra inmediatamente debajo de aquél de cuya dirección partimos. Lamentable-

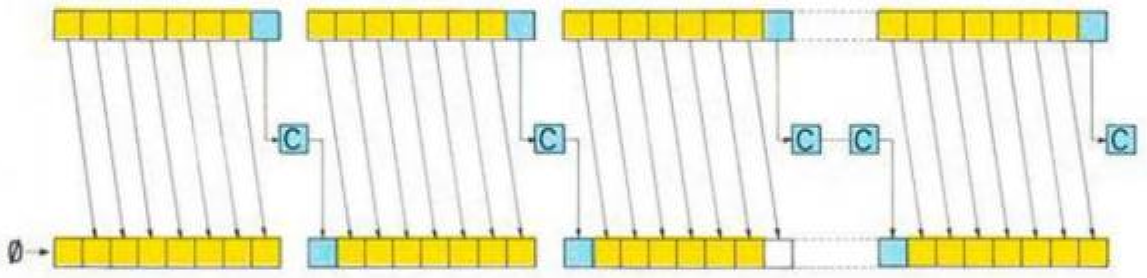


Fig. 10-6. Rotación encadenada de octetos.

teto del scan; este bucle utiliza la instrucción «RR (HL)» para ir rotando cada octeto del scan; recuerde que, con esta instrucción, el bit de la derecha de cada octeto pasa al indicador de acarreo y el anterior contenido de este indicador pasa al bit de la izquierda del octeto, de esta forma, encadenamos las rotaciones de los 32 octetos (ver Figura 1Ø-6).

Al final del bucle, los 32 octetos del scan habrán rotado como si de uno solo se tratase y tendremos, en el indicador de acarreo, el contenido del bit Ø del último octeto. Si este bit es «1», tendremos que poner a «1» el bit 7 del primer octeto del scan. Para que no nos entre «morralla» por el lado izquierdo del scan, nos aseguraremos que el indicador de acarreo está a «Ø» antes de entrar el bucle.

Este bucle que hemos explicado («BUC-3») se encuentra dentro de otro («BUC-4») que se repite 192 veces para los 192 scans. Vayamos viendo el listado de la rutina para rotar a la derecha. La numeración de las líneas es la misma que la del programa fuente del Prólogo:

```
===========================
240          LD     HL,16384
```

```
250          LD     C,192
===========================
260 BUC_4    LD     B,32
270          AND    A
===========================
280 BUC_3    RR     (HL)
290          INC    HL
300          DJNZ   BUC_3
310          JR     NC,NOCA_2
===========================
320          LD     (VAR),HL
330          LD     IX,(VAR)
340          SET    7,(IX-32)
===========================
350 NOCA_2   DEC    C
360          JR     NZ,BUC_4
370          RET
380 VAR      EQU    23728
===========================
```

Hemos separado el listado en 5 bloques para estudiarlo mejor. El primer bloque está compuesto por las líneas 24Ø y 25Ø; la primera inicializa el valor del puntero «HL» al primer octeto de la pantalla y la segunda fija el número de iteraciones que tendrá el bucle «BUC-4».

En el segundo bloque, se fija el número de iteraciones del bucle «BUC-3» y se pone a «Ø» el indicador de acarreo. El tercer bloque contiene el bucle que rota a la derecha un scan completo de la forma que se ve en la Figura 1Ø-6. En la línea 31Ø comprobamos si el bit que ha pasado al indicador de acarreo es cero, si es así, saltamos a «NOCA-2», si no, transferimos a «IX» porque la instrucción «SET 7, (HL-32)» no existe; por otro lado, la transferencia de «HL» a «IX» no podemos hacerla de forma directa, ya que no disponemos de una instrucción como «LD IX,HL» por lo que tenemos que recurrir al empleo de una variable intermedia que hemos denominado «VAR» y está situada en 23728 que es una dirección de las variables del Sistema que éste no utiliza. El último bloque cierra el bucle «BUC-4» y retorna cuando éste ya se ha ejecutado 192 veces.

En algunas aplicaciones, puede interesar que la parte de pantalla que va desapareciendo por la derecha, no aparezca por la izquierda, sino que se pierda definitivamente, en ese caso, bastará con suprimir las líneas 31Ø, 32Ø, 33Ø y 34Ø. Aunque sería más elegante utilizar un «flag» de forma que, cuando el flag estuviera a «1», la pantalla rotase en forma esférica y cuando estuviese a «Ø», se perdiese lo que se escapara por un lado. Supongamos que el flag

ejemplo en un programa que establezca un diálogo con el usuario, podría definirse un flag que indique el sexo de éste, en un momento que el programa pregunte si es hombre o mujer se activaría o no el flag en función de la respuesta; a partir de este momento siempre que el programa se dirija personalmente al usuario consultará el flag y empleará el género masculino o femenino según corresponda.

En esta misma línea el programa monitor del SPECTRUM, activa un flag cuando se presiona por primera vez la tecla CAPS LOCK, por lo tanto al ir a poner una letra consulta el flag y como está activo la pone mayúscula; al presionar una segunda vez la tecla CAPS LOCK se borrará el flag. Es decir, lo que en última instancia decide si las letras son mayúsculas o minúsculas es un flag.

Un uso más amplio sería el de indicar los atributos de cualquier dato que se tenga almacenado. Por ejemplo, en un registro de un fichero de libros podríamos indicar con flags: si es bueno o malo, si lo hemos leído o no, su género literario, etc.

Otro uso, al cual podemos denominar dinámico, sería un flag que cambia en base a una condición variable. Por ejemplo, en un programa de juego, después que el jugador ha fallado varias veces seguidas se puede activar un flag, al tenerlo activo y por las consultas correspondientes el juego baja automáticamente el nivel; una vez realizados varios aciertos seguidos desactiva el flag y por tanto subiría el nivel.

INSTRUCCIONES DE MANIPULACION DE BITS

NEMONICO	INDICADORES S	Z	x	H	x	P/V	N	C	No. DE BYTES	CICLOS MEM.	REL.
BIT b,r	x	♦	x	1	x	x	Ø	.	2	2	8
BIT b,(HL)	x	♦	x	1	x	x	Ø	.	2	3	12
BIT b,(IX+d)	x	♦	x	1	x	x	Ø	.	4	5	2Ø
BIT b,(IY+d)	x	♦	x	1	x	x	Ø	.	4	5	2Ø
SET b,r	.	.	x	.	x	.	.	.	2	2	8
SET b,(HL)	.	.	x	.	x	.	.	.	2	4	15
SET b,(IX+d)	.	.	x	.	x	.	.	.	4	6	23
SET b,(IY+d)	.	.	x	.	x	.	.	.	4	6	23
RES b,r	.	.	x	.	x	.	.	.	2	2	8
RES b,(HL)	.	.	x	.	x	.	.	.	2	4	15
RES b,(IX+d)	.	.	x	.	x	.	.	.	4	6	23
RES b,(IY+d)	.	.	x	.	x	.	.	.	4	6	23

NOTAS:

1.- Los signos tienen el siguiente significado:
"♦": El indicador cambia de valor de acuerdo con el resultado de la instrucción.
"x": El bit adquiere un estado indeterminado.
".": El indicador no es afectado por la instrucción y conserva su anterior contenido.
"Ø": El indicador se pone siempre a "cero".
"1": El indicador se pone siempre a "uno".

2.- La letra "r" indica cualquiera de los registros: "A", "B", "C", "D", "E", "H" ó "L".

Fig. 10-4. Tabla resumida de indicadores y ciclos para las instrucciones de manipulación de bits.

UTILIZACION DE LOS FLAGS

Un flag podría ser utilizado en cualquier campo, un octeto, varios octetos o un bit. Dada la condición binaria del flag y la facilidad que tiene el micro-procesador Z-8Ø de manejar bits de forma independiente, parece lo más aconsejable usar el bit como flag.

Una vez decidido que el bit «b» del octeto «o» indica la condición «c», nos limitaremos a utilizar la instrucción SET cuando lo queramos activar, la instrucción RES cuando lo queramos desactivar y la instrucción BIT asociada con un salto condicional del indicador «Z» cuando queramos analizar su estado. Esta

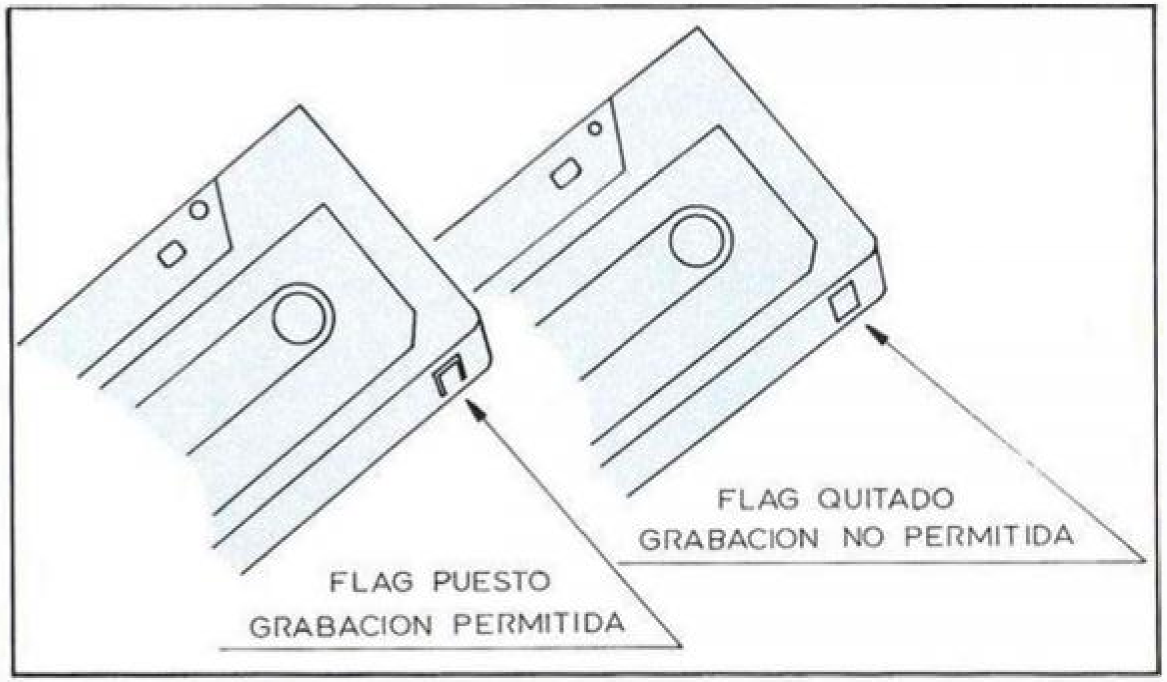


Fig. 10-5. Un ejemplo de «Flag» en la vida real.

sería la forma más inmediata y sencilla de utilizar un flag, tanto de manera estática como dinámica.

El uso de flags como atributos puede realizarse de formas más diversas. Para verlo lo ilustraremos siguiendo un ejemplo.

Supongamos que tenemos un fichero de libros en el que cada registro o ficha tiene el siguiente formato y contenido.

TITULO: 30 octetos

AUTOR: 30 octetos

TEMA: 1 octeto

- bit 7 = 1 novela
- bit 6 = 1 ensayo
- bit 5 = 1 ciencia
- bit 4 = 1 historia
- bit 3 = 1 cuento
- bit 2 = 1 poesía
- bit 1 = 1 teatro
- bit 0 = 1 otros

ATRIBUTOS: 1 octeto

- bit 7 = 1 bueno
- bit 6 = 1 interesante
- bit 5 = 1 malo
- bit 4 = 1 leído
- bit 3 = 1 para entendidos
- bit 2 = 1 para mayores
- bit 1 = 1 en inglés
- bit 0 = 1 en francés

EDITORIAL: 15 octetos

FECHA: 1 octeto

MES: 1 octeto

AÑO: 1 octeto

Podría ponerse mucha más información, pero para lo que nos ocupa nos dedicaremos principalmente a los octetos denominados TEMA y ATRIBUTOS.

En el momento que estamos creando el fichero con los libros que se tienen o cada vez que se incluye un nuevo libro, se irán llenando los campos y activando los bits del octeto TEMA y ATRIBUTOS, algunos de los atributos se llenarán después de haber leído el libro. Toda esta información, que ocupa tan poco espacio, se actualizaría con las instrucciones SET dado que el estado inicial es cero.

Supongamos que queremos ahora localizar una novela que no hayamos leído. El problema se limitaría a ir buscando con la instrucción BIT en el octeto TEMA de cada libro, si el bit 7 está activo. Cada vez que localicemos uno, miraremos con la misma instrucción el bit 4 del octeto ATRIBUTOS de ese libro; una vez localizado uno que no esté activo sacaremos a pantalla el TITULO y AUTOR. Para localizar este libro sólo hemos utilizado la instrucción BIT de manipulación de bits.

Una nueva utilización sería el caso de que un amigo nos pide un libro prestado y nos pone las siguientes condiciones: que trate sobre ciencia, de la cual sabe poco y le gustaría que estuviera escrito en inglés, por otra parte nosotros preferimos dejarle uno que ya hayamos leído. Para localizar los libros de ciencia emplearíamos el mismo método que en el ejemplo anterior, pero éste resultaría engorroso para analizar el octeto de ATRIBUTOS. Un método bueno sería preparar un octeto con los bits que queremos analizar activos y hacer una operación lógica AND con el octeto ATRIBUTOS. Este octeto con los bits activos se llama «máscara». Para nuestro ejemplo la máscara sería la siguiente:

10011010

con la que analizaremos los atributos BUENO, LEIDO, PARA ENTENDIDOS y EN INGLES; pero como nuestro amigo no tiene mucha idea del tema tendremos que descartar todos los que tengan el atributo PARA ENTENDIDOS, esto se puede solucionar comparando el resultado de la instrucción AND con el valor:

10010010

con lo que sólo nos quedaríamos con los libros de CIENCIA, que son BUENOS, BASICOS, en INGLES y LEIDOS.

Resumiendo la secuencia de instrucciones sería:

Primero: BIT para localizar los libros de tema CIENCIA.

Segundo: una vez localizado uno, aplicar al octeto ATRIBUTOS un operador AND con una máscara que tenga unos en los flags que queremos analizar.

Tercero: aplicar al resultado una resta (CP) con los valores de los flags que queremos que estén activos.

Cuarto: cuando esta instrucción nos dé como resultado cero, sacar a la pantalla el TITULO Y AUTOR pues ya hemos localizado un libro.

Los flags en informática están por todas partes. Dentro de las variables del Sistema en el Spectrum existen tres octetos con diferentes flags para control del programa monitor éstos son: «FLAGS», «TV-FLAG» y «FLAGS2». También los indicadores de condición son flags con un uso dinámico.

Como habrá comprendido el lector, la utilización de flags es ilimitada. Es muy recomendable utilizarlos al máximo pues es una información que ocupa muy poca memoria y su acceso resulta sencillo. Ya señalamos al principio de este curso que la primera aplicación del microprocesador Z-80 fue industrial, seguramente éste es el motivo por el cual trae incorporadas estas instrucciones de manipulación de bits y no trae otras más usuales como son la multiplicación, división o desplazamientos de más de un bit. En cualquier caso, la obligación de toda persona que pretende hacer un programa es usar al máximo la potencia del procesador y ahorrar la memoria del ordenador que éste está empleando.

Ejemplos

En el Prólogo de este curso (número 52 de MICROHOBBY), publicamos una rutina de ejemplo que servía para mover la pantalla a izquierda y derecha, pixel a pixel. La única finalidad de esa rutina era mostrar al lector lo que se podía conseguir empleando el Código Máquina. En ese momento, no se explicó su funcionamiento dado que, se supone, que el lector carecía del conocimiento necesario de Código Máquina para comprenderlo. Ahora, por fin, ha llegado el momento de explicar esta rutina y desarrollar otra que permita hacer lo mismo, pero en el sentido vertical, como ya prometíamos en el Prólogo.

Empezaremos por explicar la rutina de «SCROLL» lateral. Suponemos que casi todos los lectores teclearían el programa que manejaba esta rutina y se quedarían impacientes esperando la prometida rutina de «SCROLL» vertical. Posiblemente, algunos ya hayan intentado profundizar en su funcionamiento. Todas las instrucciones que se utilizan han sido vistas ya; sólo faltaban por ver las de las líneas 170 y 340 (SET 0, (IX + 32) y SET 7, (IX—32) respectivamente), precisamente, las que hemos visto en este capítulo.

Suponemos que las rutinas no existen y empezaremos por el principio, es decir, por plantearnos el problema a resolver: Se trata de desplazar toda la pantalla un pixel a la derecha, teniendo en cuenta que los bits que se escapen por la derecha de cada scan deberán entrar por la izquierda del mismo, para conseguir el efecto de una pantalla circular. La solución más sencilla consiste en ir desplazando, uno a uno, los 192 scans que componen la pantalla empezando por el primero y acabando por el último.

Para cada scan, utilizaremos un bucle de 32 iteraciones en el que entraremos con «HL» apuntando al primer oc-

CURSO DE BASIC + MICROORDENADORES

prácticas con...

Microordenador ZX SPECTRUM



Microordenador COMMODORE



Para saber cómo hablar con los ordenadores

El Curso CEAC a Distancia, BASIC + Microordenadores, le va a introducir paso a paso, con un cuidado método, en uno de los temas más apasionantes de nuestros días: **la programación de ordenadores.**
Al aprender PRACTICANDO desde un principio a programar BASIC, lenguaje diseñado especialmente para dar los primeros pasos en programación, estará sentando las bases para el estudio de cualquier otro lenguaje de alto nivel.

Curso CEAC de BASIC + Microordenadores: un diálogo permanente con el ordenador.

Otros Cursos:

- Introducción a la Informática
- Electrónica (con experimentos)
- Contabilidad
- Fotografía
- Curso de Video
- Decoración

CEAC

CENTRO DE ENSEÑANZA A DISTANCIA AUTORIZADO POR EL MINISTERIO DE EDUCACION Y CIENCIA N.º 8039185
(BOLETIN OFICIAL DEL ESTADO 3-6-83)
Aragón, 472 (Dpto. M-XN) 08013 Barcelona
Tel.: (93) 245 33 06

o llame... (93) 245 33 06 de Barcelona

CEAC PREPARANDO FUTURO

ESTAS ENSEÑANZAS SE AJUSTAN AL ART. 35 DEL DECRETO 707/1976 Y A LA ORDEN MINISTERIAL DE 5/2/1979

Actúe ahora en su propio beneficio y pídanos información.

GRATUITAMENTE

Sí, deseo recibir a la mayor brevedad posible información sobre el Curso de: ______________________

____

Nombre y apellidos ______________________ Edad ____

Domicilio ______________________

______________ Nº ____ Piso ____ Pta. ____ Tel. ______________

C. Postal ________ Población ______________________

________________ Provincia ______________________

Profesión ______________________

CEAC. Aragón, 472 (Dpto. M-XN | 08013 Barcelona

889

INTERCAMBIO DE INFORMACION BASIC-CODIGO MAQUINA

Ricardo SERRAL WIGGE



Javier Igual

Los ordenadores domésticos suelen ser tan espartanos que nos vemos tentados a aumentar su capacidad, por ejemplo, programando nuevos comandos. En este artículo se explica cómo se pueden enlazar el código máquina y el Basic del Spectrum. Las dos nuevas funciones que se dan como ejemplo se pueden utilizar en nuestros programas en Basic sin necesidad de conocer el código máquina del Z-80.

Normalmente si se quieren transferir parámetros (variables, expresiones, valores, funciones, etc.) desde un programa en Basic a una subrutina en código máquina, se suelen «pokear» en un área de memoria seleccionada (por ejemplo, el «buffer» de la impresora) de modo que la rutina en código máquina sabe que puede encontrar esa información en estos lugares determinados.

Este método es perfectamente válido, si la subrutina en código máquina se llama pocas veces desde el Basic y se intercambian unos pocos parámetros, como por ejemplo, puntuaciones máximas y niveles de dificultad en videojuegos.

No obstante, si lo que se pretende es ampliar el Basic con nuevas funciones y sentencias usando el método anterior, los programas en Basic pueden parecerle ilegibles a la gente que no conoce el código máquina, debido al gran número de PEEKs y POKEs que éste implica y además, tiene el inconveniente de que estas aplicaciones son difíciles de adaptar a otros programas en Basic, distintos al programa para el que fueron diseñados.

La forma más universal de enfocar este problema es diseñar la ampliación como módulo independiente y enlazarla con los programas que la han de utilizar mediante el paso de parámetros, tal y como se hace en algunos lenguajes de alto nivel como el Pascal. Lamentablemente el Basic del Spectrum no prevé esta eventualidad.

Otros intérpretes de Basic, como el del ORIC que tiene un Basic estructurado, resuelven este inconveniente que tiene el Spectrum con símbolos reservados como "!" y "&(x)" donde:

"!" permite ejecutar una sentencia escrita por el usuario en código máquina con tantos argumentos como se quiera.

"&(x)" llama una función en código máquina definida por el usuario que tiene un argumento numérico.

En el Basic del ZX-Spectrum hay dos tipos básicos de datos: cadenas y números (coma flotante o enteros).

La función USR, tal y como está implementada, ya nos permite devolver un resultado numérico entero desde una subrutina en código máquina al Basic, si éste se deja en el registro doble BC. Pero no es tan fácil pasar números reales ni cadenas al Basic.

Sin embargo, examinando la ROM del Spectrum podemos darnos cuenta de que es fácil utilizar la sentencia DEF FN para definir nuevas funciones o comandos en código máquina, que precisen un número indefinido de argumentos de cualquier tipo (expresiones de cadena o numéricas), sin interferir con las variables normales del Basic y sin que haga falta ningún PEEK o POKE para llamarlas.

Los valores de los argumentos de una función definida por el usuario FN, no se almacenan como variables, sino que se insertan en el mismo programa en Basic como parámetros numéricos invisibles justo después del nombre de cada argumento en la sentencia DEF FN correspondiente.

Escribiendo el ejemplo del programa 1, podremos ver estos parámetros invisibles.

```
 10 DEF FN I(X$,I$,P)=PI
 20 PRINT FN I("HOLA","LA",2)
 30 FOR N=23755 TO 23850: PRINT
N;" = ";PEEK N,CHR$ PEEK N AND
PEEK N>31: NEXT N
```

Después del último carácter del nombre del argumento viene un carácter CHR$(14) seguido por cinco bytes de datos. Si el argumento es numérico, los cinco bytes contendrán un número de coma flotante o un entero. Si el argumento es una cadena o una matriz de caracteres los datos serán un juego de parámetros de cadena como sigue:

1 byte para el tipo de cadena (0 = matriz de caracteres, 1 = = cadena, otro = literal).
2 bytes para la dirección de comienzo de la cadena.
2 bytes para la longitud de la cadena.



LISTADO I. DESENSAMBLE DE LA RUTINA INSTRING

```
*C-
        ORG  #FF58
INSTR   PUSH DE
        PUSH HL
        PUSH IX
        LD   IX,(23563);IX=DEF-ADD
        LD   A,(IX+6)
        OR   (IX+7)
        JR   Z,NT_FND ;Sal y haz BC=0000 si X$=""
        LD   E,(IX+4)
        LD   D,(IX+5) ;DE=direccion de X$
        LD   L,(IX+6)
        LD   H,(IX+7) ;HL=longitud de x$
        ADD  HL,DE
        LD   (23728),HL;(23728)=indicador final de X$
        LD   L,(IX+22)
        LD   H,(IX+23);HL=P
        DEC  HL
        ADD  HL,DE
N_MTCH  LD   E,(IX+13)
        LD   D,(IX+14)
        DEC  DE;DE=direccion de I$ - 1
        LD   C,(IX+15)
        LD   B,(IX+16);BC=longitud de X$
        NOP  ;busca I$ dentro de X$
MATCH   INC  DE
        LD   A,(DE)
        CPI
        EX   AF,AF'
        PUSH HL
        PUSH DE
        AND  A
        LD   DE,(23728);DE=indicador final de X$
        SBC  HL,DE
        POP  DE
        POP  HL
        JR   Z,END_2;salta si ultimo caracter de X$
        LD   A,B
        OR   C
        JR   Z,END_1;salta si ultimo caracter de I$
        EX   AF,AF'
        JR   Z,MATCH;salta si (HL)=(DE)
        JR   N_MTCH
        DEFS 7
END_1   EX   AF,AF'
        JR   NZ,N_MTCH;salta si (HL)<>(DE)
        JR   FOUND
END_2   LD   A,B
        OR   C
        JR   NZ,NT_FND;salta si BC<>0000
        EX   AF,AF'
        JR   NZ,NT_FND;salta si (HL)<>(DE)
        JR   FOUND
NT_FND  LD   BC,#0000
        JR   EXIT
FOUND   LD   E,(IX+4)
        LD   D,(IX+5) ;DE=direccion de X$
        AND  A
        SBC  HL,DE
        LD   E,(IX+15)
        LD   D,(IX+16);DE= longitud de I$
        AND  A
        SBC  HL,DE
        INC  HL
        LD   B,H
        LD   C,L ;BC=Posicion de I$ en X$
EXIT    POP  IX
        POP  HL
        POP  DE
        RET
```

Esto sería totalmente inútil si no pudiéramos localizar la posición de esta información en cualquier programa Basic, independientemente del sitio en donde se haga la definición de la función. Afortunadamente la variable del sistema DEF-ADD *durante* la ejecución de una función FN contiene la dirección del primer argumento de la función en la sentencia DEF FN (el primer carácter despues de '(')). En otro caso su valor es cero.

Podemos ver todo esto con más claridad en la figura 1.

Como ejemplo de aplicación del método para una función con argumentos numéricos y alfanuméricos que devuelve un resultado numérico, el listado 1 muestra una función INSTR (X$, I$, P) tal y como la incorporan algunos intérpretes de Microsoft.

La función INSTRING busca la posición de la cadena I$ dentro de la cadena X$ a partir del pésimo carácter de X$.

```
                    ⎧ POS si I$<X$ (P TO)
INSTR (X$, I$, P)=  ⎨
                    ⎩ 0 { si no I$<X$ (P TO)
                          si X$=""
```

Esta función es sumamente útil para escribir programas de tratamiento de textos, ensambladores, analizadores sintácticos, juegos de aventuras con diálogos, etc...

Cargando el listado 1 con un monitor hexadecimal o un ensamblador a partir de la dirección USR "A", sustituyendo PI en el ejemplo 1 por USR USR "A" y borrando la línea 100, se puede ver cómo trabaja la función INSTR (POS se devuelve en el registro doble BC). Es posible sustituir los argumentos de la función FN I de la línea 20 por variables de cadena, cadenas troceadas, literales e incluso elementos (campos) de matrices de caracteres.

La siguiente subrutina equivalente demuestra el algoritmo utilizado, aunque es mucho más lenta.

```
8000 LET POS=0
8010 FOR N=P TO LEN X$-LEN I$+1
8020 IF X$(N TO N+LEN I$-1)<>I$
THEN NEXT N: RETURN
8030 LET POS=N: RETURN
```

El listado 1 explica una forma de recoger desde el código máquina los parámetros suministrados por el Basic al ejecutar la función FN I.

```
23755       0          } numero de linea
23756      10          }
23757      LL          } longitud de la linea
23758      LH          }
23759     206  DEF FN  Token
23760      73       I
23761      40       )
23762  0   88       X
23763  1   36       $
23764  2   14          marca parametros num.
23765  3    T          tipo de cadena
23766  4   AL          } direccion de
23767  5   AH          } comienzo de X$
23768  6   LL          } longitud
23769  7   LH          } de X$
23770  8   44       ,
23771  9   73       I
23772 10   36       $
23773 11   14          marca parametros num.
23774 12    T          tipo de cadena
23775 13   AL          } direccion de
23776 14   AH          } comienzo de I$
23777 15   LL          } longitud de I$
23778 16   LH          }
23779 17   44       ,
23780 18   80       P
23781 19   14          marca parametros num.
23782 20    0
23783 21    S          Byte de signo 0=+ ; 255=-
23784 22   NL          LSB } numero
23785 23   NH          MSB } entero
23786 24    0
23787      41       )
23788      61       =
```

Figura 1.

LISTADO 2. DESENSAMBLE DE LA RUTINA HEX$TRING

```
1 *C-
10          ORG  #FEE0
20 HEXN     PUSH IX
30          LD   IX,(23563) ;IX=DEF-ADD
40          LD   HL,23300 ;buffer de la impresora
50          LD   E,(IX+4)
60          LD   D,(IX+5) ;DE=numero decimal entero
70          XOR  A
80          CP   D ;DE>255 ? conmuta entre Byte y Palabra
90          LD   BC,#0002
100         JR   Z,BYTE
110 WORD    LD   C,4
120         LD   A,D
130         CALL BINHEX ;convierte MSB
140 BYTE    LD   A,E
150         CALL BINHEX ;convierte LSB
160 STACK   LD   DE,(23300) ;apunta con DE al literal hexadecimal
170         LD   HL,(23653) ;HL=STK-END
180         DEC  HL ;apunta al quinto Byte de 'LAST VALUE' en la pila
190 ;del calculador
200         LD   (HL),B
210         DEC  HL
220         LD   (HL),C ;fija la longitud del literal hexadecimal
230         DEC  HL
240         LD   (HL),D
250         DEC  HL
260         LD   (HL),E ;fija la direccion de comienzo del literal
270         POP  IX
280         RET  ;vuelve con BC<>0000
290 BINHEX  PUSH BC ;entra con A conteniendo el Byte a convertir
300         LD   C,A
310         SRL  C
320         SRL  C
330         SRL  C
340         SRL  C ;divide por 16
350         LD   B,C ;guarda MSD en B
360         SLA  C
370         SLA  C
380         SLA  C
390         SLA  C ;C=16*INT(A/16)
400         SUB  C ;A=A mod 16
410         LD   C,A ;guarda LSD en C
420         LD   A,B
430         CALL SR ;CHR$ (LSD)
440         LD   A,C
450         CALL SR ;CHR$ (MSD)
460         POP  BC
470         RET
480 ;
490 SR      ADD  A,#30 ;convierte numeros al codigo ASCII
500         CP   #3A
510         JR   C,WRITE
520         ADD  A,7 ;convierte letras al codigo ASCII
530 WRITE   LD   (HL),A
540         INC  HL
550         RET
```

Vamos a dar ahora un paso más. Si queremos devolver una cadena de caracteres desde el código máquina al Basic, tenemos que usar un truco para engañar al intérprete.

Definimos una función de la forma: DEF FN H$ (lista de argumentos)= = "PRUEBA" AND USR (dirección de entrada de la subrutina en código máquina).

A primera vista parece que una llamada a la función FN H$ nos dará como resultado la cadena de caracteres "PRUEBA", si USR (dirección) devuelve un entero distinto de cero, o en otro caso la cadena vacía " ". Dado que la prioridad de USR es mayor que la de AND para el evaluador de expresiones del Spectrum, podemos asegurar que el primer argumento de la función AND (en este caso los parámetros de cadena de "PRUEBA"), se encontrará como último elemento de la pila del calculador y que los códigos de operación de ambas funciones estarán en la pila de la máquina durante la ejecución de la función USR.

Al entrar en nuestra subrutina en código máquina, la variable del sistema STK-END apuntará al primer byte después del literal "PRUEBA". Sólo tenemos que alterar dichos parámetros haciéndolos apuntar a una cadena que hemos construido en algún lugar de la memoria (p. ej.: el buffer de la impresora). Como el evaluador de expresiones ya espera un resultado alfanumérico, no tenemos que alterar ningún "flag". No obstante, tenemos que observar que al regresar del código máquina al Basic el registro doble BC ha de contener un número entero no nulo.

Si se quiere implementar una función que devuelva un resultado en coma flotante podemos usar el mismo método, pero obligando al evaluador de expresiones a asumir un resultado numérico mediante una expresión del tipo

PI AND USR (dirección de entrada de la subrutina en código máquina).

El programa del listado 2, que convierte un número decimal en una cadena de caracteres hexadecimal, puede servir para ilustrar este método. La rutina no es relocalizable.

Para cargarlo en la memoria del ordenador se procede de forma análoga como con el listado 1, escribiendo ahora para el Spectrum de 48K:

```
  10 DEF FN H$(X)="HEX$" AND USR
65248
  20 PRINT FN H$(cualquier enter
o de 0 a 65535)
```

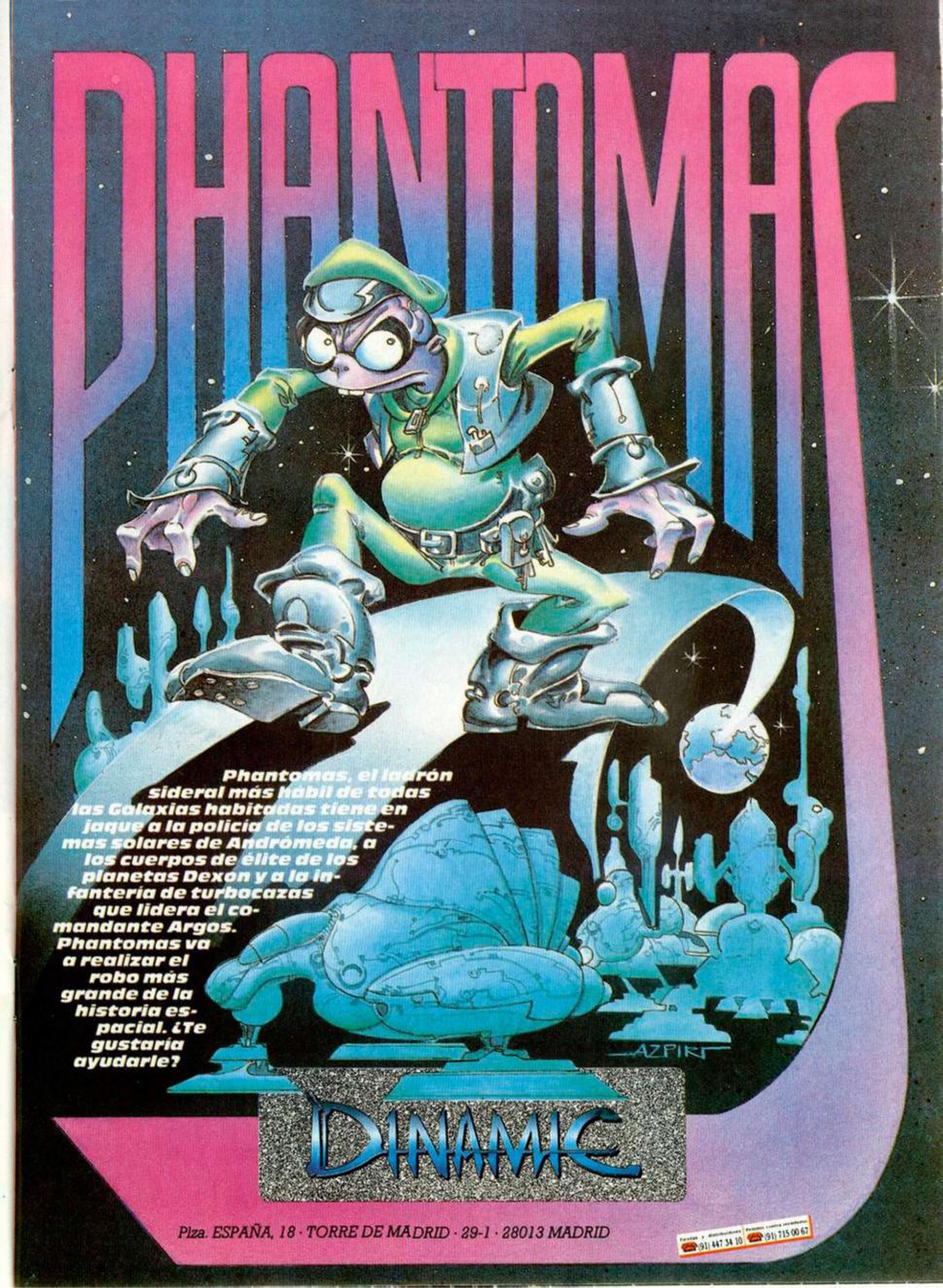
Así una vez arrancado el programa, si escribimos por ejemplo en modo directo PRINT FN H$(4095) obtendríamos como resultado en pantalla el número hexadecimal OFFF.

Para evitar el uso de un ensamblador o de un monitor hexadecimal, ya que poca gente dispone de estos programas, se puede teclear el programa en Basic del listado 3, que genera y relocaliza el código máquina para las funciones INSTRING Y HEX$ y sirve además como ejemplo para la integración de las mismas en un programa en Basic.

El método descrito en este artículo supone un poco más de trabajo a la hora de diseñar las subrutinas en código máquina, pero a cambio de esto hace que nuestros programas en Basic sean más flexibles y transparentes.

LISTADO 3

```
  10 REM INSTRING /HEX$ Demo
  20 REM © Ricardo Serral Wigge
  30 DEF FN I(X$,I$,P)=USR comzi
nstr
  40 DEF FN H$(X)="HEX$" AND USR
 comzhex
  50 CLS
 100 INPUT "INSTRING o HEX$  [I,
H] ", LINE A$
 110 IF A$="i" OR A$="I" THEN GO
 TO 800
 120 IF A$="h" OR A$="H" THEN GO
 TO 900
 130 GO TO 100
 400 REM INSTRING demo
 410 INPUT "Introduce una cadena
 ",x$: PRINT "Cadena:""""";x$;"
"""
 420 INPUT "Cadena a buscar ",i$
: PRINT "Cadena a buscar:""""";
i$;""""
 430 INPUT "A partir de la posic
r+132: READ a: POKE n,a: NEXT n:
 GO TO 400
 900 REM generar HEX$
 910 INPUT "Direccion de comienz
o: ",comzhex
 920 POKE 23300,comzhex-256*INT
(comzhex/256): POKE 23301,INT (c
omzhex/256): CLEAR comzhex-1: LE
T comzhex=PEEK 23300+256*PEEK 23
301
 950 RESTORE 2000
 960 LET binhexl=(comzhex+49)-25
6*INT ((comzhex+49)/256): LET bi
nhexh=INT ((comzhex+49)/256)
 970 LET srl=(comzhex+80)-256*IN
T ((comzhex+80)/256): LET srh=IN
T ((comzhex+80)/256)
 980 FOR n=comzhex TO comzhex+90
: READ a: POKE n,a: NEXT n: GO T
O 500
1000 DATA 213,229,221,229,221,42
,11,92
ion: ",p: PRINT "A partir de la p
osicion :";p""
 440 LET pos=FN i(x$,i$,p)
 450 PRINT "se encuentra a parti
r de la posicion ";pos;" de la p
rimera cadena"""
 460 GO TO 400
 500 REM HEX$ demo
 510 INPUT "Introduce un numero
decimal: ";x: PRINT x;"d = ";FN h
$(x);"h """
 520 GO TO 500
 800 REM generar INSTRING
 810 INPUT "Direccion de comienz
o: ",comzinstr
 820 POKE 23300,comzinstr-256*IN
T (comzinstr/256): POKE 23301,IN
T (comzinstr/256): CLEAR comzins
tr-1: LET comzinstr=PEEK 23300+2
56*PEEK 23301
 850 RESTORE 1000
 860 FOR n=comzinstr TO comzinst
1010 DATA 221,126,6,221,182,7,40
,86
1020 DATA 221,94,4,221,86,5,221,
110
1030 DATA 6,221,102,7,25,34,176,
92
1040 DATA 221,110,22,221,102,23,
43,25
1050 DATA 221,94,13,221,86,14,27
,221
1060 DATA 78,15,221,70,16,0,19,2
6
1070 DATA 237,161,8,229,213,167,
237,91
1080 DATA 176,92,237,82,209,225,
40,21
1090 DATA 120,177,40,12,8,40,231
,24
1100 DATA 215,32,205,24,14,120,1
77,32
1110 DATA 8,32,205,24,14,120,177
,32
1010 DATA 221,126,6,221,182,7,40
,86
1020 DATA 221,94,4,221,86,5,221,
110
1030 DATA 6,221,102,7,25,34,176,
92
1040 DATA 221,110,22,221,102,23,
43,25
1050 DATA 221,94,13,221,86,14,27
,221
1060 DATA 78,15,221,70,16,0,19,2
6
1070 DATA 237,161,8,229,213,167,
237,91
1080 DATA 176,92,237,82,209,225,
40,21
1090 DATA 120,177,40,12,8,40,231
,24
1100 DATA 215,32,205,24,14,120,1
77,32
1110 DATA 8,32,205,24,14,120,177
,32
1120 DATA 5,8,32,2,24,5,1,0
1130 DATA 0,24,21,221,94,4,221,8
6
1140 DATA 5,167,237,82,221,94,15
,221
1150 DATA 86,16,167,237,82,35,68
,77
1160 DATA 221,225,225,209,201
2000 DATA 221,229,221,42,11,92,3
3,4
2010 DATA 91,221,94,4,221,86,5,1
75
2020 DATA 186,1,2,0,40,6,14,4
2030 DATA 122,205,binhexl,binhex
h,123,205,binhexl,binhexh,17,4
2040 DATA 91,42,101,92,43,112,43
,113
2050 DATA 43,114,43,115,221,225,
201,197
2060 DATA 79,203,57,203,57,203,5
7,203
2070 DATA 57,65,203,33,203,33,20
3,33
2080 DATA 203,33,145,79,120,205,
srl,srh
2090 DATA 121,205,srl,srh,193,20
1,198,48
2100 DATA 254,58,56,2,198,7,119,
35
2110 DATA 201
9998 REM
9999 SAVE "INS/HEXD" LINE 1
```


PHANTOMAS
Phantomas, el ladrón sideral más hábil de todas las Galaxias habitadas tiene en jaque a la policía de los sistemas solares de Andrómeda, a los cuerpos de élite de los planetas Dexon y a la infantería de turbocazas que lidera el comandante Argos. Phantomas va a realizar el robo más grande de la historia espacial. ¿Te gustaría ayudarle?
AZPIRI
DINAMIC
Plza. ESPAÑA, 18 · TORRE DE MADRID · 29-1 · 28013 MADRID
(91) 447 34 10
(91) 715 00 67

LOS JUSTICIEROS DEL SOFTWARE

MOVIE

Esta videoaventura de Imagine parece contemplar todo lo necesario para ser considerada «excepcional», pues éste ha sido el veredicto de nuestros justicieros, como podréis comprobar.

«Pantallas Monocolor»

● POSITIVO
Es un juego brillante por su originalidad y sus magníficos gráficos. Altamente adictivo y de dificultad elevada. La presentación es sencilla y original.

● NEGATIVO
Pocos efectos sonoros. Las pantallas son monocolor. Movimiento algo rígido y el control por teclas algo confuso por la perspectiva.

PUNTUACION: 9
Gabriel Martí Fuentes

«Sensacional»

● POSITIVO
Es uno de los mejores que he visto. Es original, tanto por la temática como por la f ma de controlar al personaje. Los gráficos son perfectos y el movimiento a pesar de no ser rápido, está bien conseguido. Es un programa muy interesante.

● NEGATIVO
El sonido es lo único negativo en este juego.

PUNTUACION: 10
Ricardo Alonso Villemur

«Escasez de sonido»

● POSITIVO
Los gráficos excepcionales. Movimiento fantástico, gran adicción y un buen número de pantallas. Tiene detalles geniales, como poder hablar con los personajes. Magistral pantalla de presentación. Una vídeo aventura con todas las de la ley.

● NEGATIVO
Tal vez lo más flojo sea el sonido ya que casi no existe durante el desarrollo, pero a pesar de esto sigue siendo magnífico.

PUNTUACIÓN: 10
David de Llodio Olaizola

«Un juego muy adictivo»

● POSITIVO
Las opciones para el manejo del juego son numerosas. Los gráficos son excepcionales con un lujo de detalles extremos. El sonido es muy bueno. El juego en sí es completísimo, combinando muy sabiamente la habilidad con la lógica y la deducción. Se trata de un juego muy adictivo para pasar largas horas de entretenimiento.

● NEGATIVO
Las pantallas son unicolores, se hace en algún momento un poco lento.

Puntuación: 10
Fco. Javier Cano Peño

«Un auténtico n.º 1»

● POSITIVO
Tiene una gran originalidad ya que podemos conversar con los personajes del juego. La gran variedad de las pantallas, todas muy bien realizadas y con gran cantidad de detalles, hacen del él un auténtico n.º 1

● NEGATIVO
Carece de sonido y el movimiento es un tanto dificultoso.

PUNTUACION: 8
José Antonio Galiana

«Gráficos bien ambientados»

● POSITIVO
Combina perfectamente las situaciones de acción con los momentos en que hay que detenerse y pensar, para seguir adelante, dejando la puerta abierta a la imaginación del jugador. Los gráficos están bastante bien y consiguen crear ambiente.

● NEGATIVO
Es inexplicable que algunos detalles que están tratados bien gráficamente no tengan sonido alguno, por ejemplo, los disparos o golpes.

PUNTUACION: 9
Pilar Arias Valcarce

«Complicación de movimiento»

● POSITIVO
Muy buenos gráficos y adictividad. Tiene gran cantidad de pequeños detalles que hacen más ameno el juego, como por ejemplo, el movimiento del péndulo del reloj y otros efectos que aparecen por el juego.

● NEGATIVO
Deja un poco que desear el movimiento de los personajes. Es un tanto complicado.

PUNTUACION: 9
Roberto Carlos Alonso

«Hará historia»

● POSITIVO
Es un programa increíble, en el que todo es destacable, gráficos, movimiento y originalidad. Podemos decir que estamos ante un juego que hará historia.

● NEGATIVO
Se podrían haber esmerado más en el sonido.

PUNTUACION: 10
Juan Markessinis



Rutinas CM en la zona de variables del Basic

LA BIBLIA DEL «HACKER» (VI)

José Manuel LAZO

Una de las formas más habituales de guardar una rutina de carga de Código Máquina dentro de un programa BASIC es hacerlo dentro de la zona de variables. De esta forma, si alguien accede al listado no podrá verlo y si ejecuta comandos del tipo RUN o CLEAR, la rutina desaparece por arte de magia.

Un bloque de CM se puede guardar perfectamente en la zona de las variables, para comprender esto es necesario saber cómo el intérprete graba un programa en Basic:

Cuando damos la orden SAVE «nombre», el SO coge la variable PROG y toma la información que la misma contiene como el primer octeto a grabar, la longitud del Basic grabado depende de lo que marque la variable E-LINE que señala el final de la zona de variables del Basic. Además, en la cabecera del programa se guarda la longitud del listado Basic dentro del bloque grabado, que puede ser igual o inferior al mismo.

De todo esto se deduce que el señor que haga la protección puede guardar perfectamente un programa en CM en la zona de variables y grabarlo junto con el programa. Una consecuencia de lo mismo puede ser que si nosotros ejecutamos un RUN se nos borran las variables, y con ello el programa en CM con el consiguiente cuelgue.

Cuando nosotros grabamos un programa con AUTO-RUN no lo hacemos de forma que se haga un RUN a la línea que marquemos sino un GOTO, una expresión que sería adecuada es: grabar un programa con AUTO-GOTO.

El código máquina cargador no tiene por qué estar necesariamente dentro del listado, al contrario, lo más sencillo para el programador es situarlo en un bloque de código grabado independientemente, aunque esto es más sencillo de desprotejer. Sólo hay que averiguar la dirección donde se carga y donde se ejecuta.

Formas de ejecutar un CM. cargador

Partimos del caso de que en el programa en Basic, que actúa como cargador, no se ve una sentencia LOAD por ninguna parte, de esto se puede deducir que los demás bloques del programa se cargan desde CM. No vamos a entrar todavía en cómo carga el programa CM, pero vamos a ver las distintas maneras que hay de llamar al mismo.

La forma más sencilla es RANDOMIZE USR dirección. Siempre que nos hayamos asegurado de que la dirección que se da sea la verdadera podemos pasar ya sin más al desensamblado, pero esta forma es poco corriente porque es muy fácil de desproteger y porque podría dar problemas si se tiene el Interface 1 conectado.

Otra forma muy común es RANDOMIZE USR (PEEK 23635 + 256* PEEK 23636) + n. Esto podría valer para arrancar un programa en CM ubicado en una línea REM al principio del listado. Si deseamos desensamblar el CM tendremos que tener en cuenta que hemos cargado el Basic en otra posición para poderlo ver, por lo que en todos los CALL y JP que haya en su interior hay que calcular la dirección sobre la que funcionan.

Si tenemos CM en la zona de variables se puede usar la forma: RANDOMIZE USR (PEEK VARS + 256*PEEK VARS + 1) + N. Esto lo que hace es una llamada a una rutina a partir de lo que contiene la variable del sistemea VARS. Cuando nos encontremos con ello habrá que tener cuidado, si estamos viendo el programa sin el Copyline, de no hacer ninguna operación que modifique las variables.

Otra manera de llamar a un programa en CM desde el Basic sin que esta llamada se note es hacer un POKE en la variable del sistema ERR SP o puntero de la dirección debido a que tiene un nivel de portección más superior a las anteriores. Vamos a estudiarla detalladamente.

Cuando se ha de presentar un informe de error el SO mira la variable ERR SP que indica la dirección del elemento de la pila de máquina que contiene, a su vez, la dirección donde se hallan las rutinas de tratamiento de errores y, acto seguido, transfiere el control del programa a esa dirección.

El SO hace esto así por varias circunstancias, pero la más importante es que en el momento en que se produce el error normalmente la pila de máquina está desequilibrada por lo que un simple RET produciría que el error no se pudiera tratar o que se colgara el ordenador.

Por esta razón, lo que se hace es guardar en esta variable de dos bytes la dirección del elemento de la pila donde se halla el retorno de error. Así, cuando el error se produce el SO mira esta dirección y hace un salto a la misma.

El programa que haga la protección puede aprovechar esto para pokear en esta variable una dirección y luego producir cualquier error, o bien por los métodos normales, BORDER 9 por ejemplo, o bien pokeando también en la variable del sistema ERR NR la cual se encarga de contener el informe de error que se ha producido.

Con esto se consigue que el SO haga directamente un salto a una rutina CM que se encuentre ubicada en la dirección contenida, a su vez, en los dos bytes hacia los que apunta la variable.

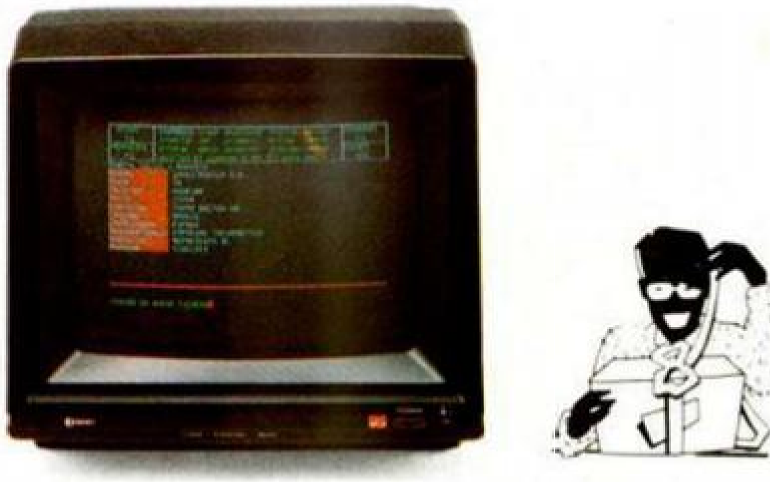
En esto se basa la protección turbo a nivel Basic, pero de eso ya hablaremos más adelante.

Por lo general la filosofía que hay que seguir a la hora de entrar en un programa Basic es muy sencilla:

- Modificar la cabecera por bytes para poderlo ubicar en otra dierección.
- Examinar el listado con el Copyline detalladamente, un POKE que se pase por alto puede ser luego una muralla infranqueable.
- Estudiar la carga de los demás bloques del programa; es posible que creamos que está superprotegido y luego sea un juego de niños.
- No dejarse engañar: muchas veces sentencias de un listado Basic pueden estar «de adorno» para confundir al «Hacker». Tampoco se debe despreciar ninguna; un simple BORDER 5 puede significar que luego se chequea la variable del sistema BORDCR para ver si está del color previsto.
- En algunos listados las literales ASCII retocadas proliferan como setas, mientras que en otros no se ha usado esta protección.
- Es muy interesante que mientras se va viendo el listado con el Copyline se vaya apuntando en un papel un listado «limpio» para que después de quitar la «paja» se pueda estudiar con más facilidad.

ASI TRABAJA EL QL HECHO PARA NOSOTROS





QL QUILL. TRATAMIENTO DE TEXTOS

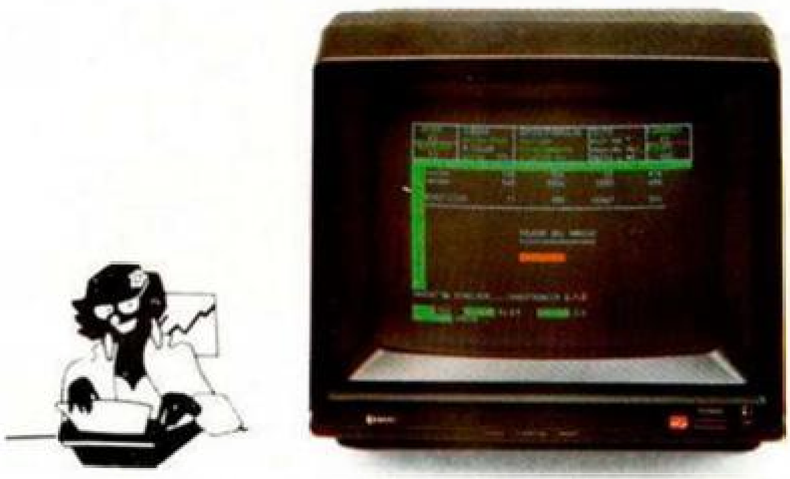
QL Quill muestra en pantalla exactamente cómo aparecerá su escrito cuando esté impreso. Olvídese de pesadas memorizaciones de comandos. Con QL Quill tiene siempre todo lo necesario en pantalla. Escriba en negrita, corte, una, fije tabuladores, márgenes, sobreescriba... después de todo esto su QL le dirá, además, cuántas palabras ha escrito.

QL ARCHIVE. BASE DE DATOS

Organizar su agenda o poner al día su fichero se convierte en un trabajo agradable. QL Archive resuelve rápidamente su problema con un sistema de archivo de gran facilidad de uso, usando un lenguaje aún más sencillo que el BASIC.

Con un mínimo de práctica puede crear desde un simple directorio a una extensa base de datos. Podrá determinar relaciones, extraer datos, reorganizarlos, hacer cálculos estadísticos y contables.

Con QL Archive podrá crear un formato personalizado para sus informes.





QL ABACUS. HOJA ELECTRONICA DE CALCULO

Este programa ha superado con mucho las tradicionales hojas de cálculo de otros ordenadores. Con QL Abacus escriba simplemente "costes de personal", o "personal", o "pers" para encontrar esa celdilla. QL Abacus también le ayuda a decidir lo que hay que hacer mediante diferentes sugerencias, dentro de una amplia gama de posibilidades. Podrá, además, crear hojas electrónicas de cálculo con más de 6.000 celdas individuales.

Podrá mostrar ventanas múltiples, variar la distancia entre columnas...

QL EASEL. GRAFICOS

Olvídese de construir tablas de valores o enfrentarse con situaciones dudosas antes de ver algún resultado. Con QL EASEL podrá desde el principio crear gráficos. Inmediatamente creará diagramas de barras, filas, líneas rellenas, sectores, barras sobrepuestas...

Todo ello simplemente pulsando una tecla.

Introduzca textos donde quiera. Dentro o cerca de su diagrama. Más aún, mueva el texto por la pantalla para ver dónde queda mejor... en el color que prefiera.

DISTRIBUIDOR EXCLUSIVO

investronica

Tomás Bretón, 60. Telf. (91) 467 82 10. Telex 23399 IYCO E. 28045 Madrid
Camp, 80. Telf. (93) 211 26 58-211 27 54. 08022 Barcelona

Para salir de un programa en Código Máquina

RESET EN CALIENTE, POR SOFTWARE

Juan Manuel PEREZ

Cuantas veces, dentro de un programa en Código Máquina, nos hubiera gustado recuperar el control perdido de nuestro ordenador sin tener que sacar la clavija de alimentación del enchufe y recuperar la valiosa información que contenía la memoria en ese momento. Esta rutina será una eficaz ayuda en este terreno.

La utilísima rutina que hoy os presentamos sirve, como ya hemos apuntado, para poder recuperar el control perdido dentro de una rutina o programa de Código Máquina. Seguramente habréis tenido la experiencia, sobre todo los que se aventuran a programar en Código Máquina, de comprobar la extrema facilidad con que puede perderse el control al ejecutar una rutina o programa en C.M., sin que el sistema operativo nos dé ningún tipo de opción a rectificar los errores cometidos.

Con esta pequeña rutina podremos, con ciertas limitaciones, solucionar este contratiempo.

Para utilizarla hay que teclear, en primer lugar, el listado 1, sirviéndonos para ello del Cargador Universal de Código Máquina publicado en el número 31 de nuestra revista. A continuación, es necesario hacer un DUMP en la dirección 40000, y salvar en cinta el Código Objeto generado, indicando 40000 como dirección de inicio y 100 como número de bytes. Es importante recordar que al cargarla desde cinta para su utilización hay que hacer LOAD""CODE 65279,100.

DATOS DE INTERES

RANDOMIZE USR 65281 - Desactiva la rutina
RANDOMIZE USR 65288 - Activa la rutina

La rutina tiene dos llamadas posibles. En una de ellas (RANDOMIZE USR 65281) se carga el registro «I» con el valor 63 (3F en hexadecimal) y se selecciona el modo 1 de interrupción. De esta manera es el sistema operativo el que lleva el control de la interrupción, dirigiéndola hacia la rutina de la ROM RST 138, que es la encargada de la lectura del teclado. La otra llamada (RANDOMIZE USR 65288) es donde se inicializa el modo 2 de interrupciones, dirigiendo el vector de interrupción hacia la rutina que hemos creado.

El siguiente pasao es leer el teclado, por si están pulsadas las teclas de control, es decir, las teclas SPACE y SYMBOL SHIFT simultáneamente. De ser así se ejecuta la rutina y si no, sigue el programa normalmente.

La rutina de Reset propiamente dicha se ubica a partir de la dirección 65317 y contiene los siguientes pasos:

Se limpia el «stack».

Se actualiza el valor de la variable «ERR-SP».

Limpia la parte inferior de la pantalla.

Especifica el cursor en modo «K».

Anula la posibilidad de que se esté ejecutando un INPUT.

Imprime el mensaje y salta a la rutina de control de la ROM.

Hay varios casos en que la rutina no actúa. Puesto que funciona por interrupciones, no operará cuando éstas estén deshabilitadas.

Tampoco funciona en el caso de la ejecución del comando BEEP ni de LOAD o SAVE. Estos comandos desactivan las interrupciones. Tampoco lo hará, por último, después de un NEW puesto que este comando repone las interrupciones al modo 1.

La rutina no se puede utilizar en un Spectrum de 16 K debido a que el registro «I» debe asumir valores que afectan gravemente al archivo de presentación visual, es decir, a la propia pantalla.

Por último, señalar que la rutina no es reubicable, por lo que no podrá utilizarse con ningún programa que resida en las mismas posiciones de memoria que ella ocupa.

Línea	Datos	Control
1	25FF3E3FED47ED56C93E	1311
2	FEED47ED5EC9160A0A4D	1213
3	4943524F484F42425916	695
4	0B0C52455345542AF53E	759
5	7FDBFEE6022804F1FFED	1609
6	4D3E7FDBFEE60120F42A	1288
7	B25C2BF92B2B223D5C3E	897
8	02CD0116CD6B0D3E0032	667
9	715C3E02CD0116110FFF	784
10	011600CD3C20FBC3A912	953

Desensamble de la rutina

```
10         ORG  65279
20         DEFW 65317
30         LD   A,#3F
40         LD   I,A
50         IM   1
60         RET
70         LD   A,#FE
80         LD   I,A
90         IM   2
100        RET
110        DEFB #16,#0A,#0A
120        DEFB "M","I","C"
130        DEFB "R","O","H"
140        DEFB "O","B","B"
150        DEFB "Y",#16,#0B
160        DEFB #0C,"R","E"
170        DEFB "S","E","T"
180        DEFB "*"
190        PUSH AF
200        LD   A,#7F
210        IN   A,(#FE)
220        AND  #02
230        JR   Z,BUCLE
240 BUCLE1 POP  AF
250        RST  #38
260        RETI
270 BUCLE  LD   A,#7F
280        IN   A,(#FE)
290        AND  #01
300        JR   NZ,BUCLE1
310        LD   HL,(#5CB2)
320        DEC  HL
330        LD   SP,HL
340        DEC  HL
350        DEC  HL
360        LD   (#5C3D),HL
370        LD   A,#02
380        CALL #1601
390        CALL #0D6B
400        LD   A,#00
410        LD   (#5C71),A
420        LD   A,#02
430        CALL #1601
440        LD   DE,#FF0F
450        LD   BC,#0016
460        CALL #203C
470        EI
480        JP   #12A9
```

MICRO Manía

Sólo para adictos

THREE WEEKS IN PARADISE

Esta vez seremos breves, que ya hemos hablado bastante de este programa. Simplemente, para el que desee obtener vidas infinitas sin tener que realizar las peripecias de la semana pasada, le bastará con teclear el siguiente Poke: POKE 500027, 201

El culpable ha sido José Aguilar, de Altea.

STARQUAKE

Adolfo López nos ha pasado un POKE casi maravilloso. Y decimos casi porque como él mismo nos cuenta «sólo da unas cuantas vidas». Sin embargo, también nos comunica que está trabajando incesantemente para conseguir redondearlo del todo.

POKE 25414, X

donde X es un número menor que 99.

Por el momento, nos asegura que estas vidas nos serán suficientes para llegar hasta el final del juego.



SWEEVO'S WORLD

Este cargador que nos ha hecho llegar Alberto Rodríguez desde Madrid, no es casi maravilloso, es completamente maravilloso.

Simplemente deberéis teclear las siguientes instrucciones, ejecutarlas con RUN (ENTER) y poner la cinta original en el cassette.



```
5 CLEAR 24799:PRINT "Sweevo's World"
10LOAD" "SCREEN$: LOAD" "CODE: LOAD" "CODE: POKE 33219, 0:RANDOMIZE USR 24800
```

CONSULTORIO

Los "piratas"

He leído en su revista número 70 que en el «Rastro» de Madrid se requisaron 11.000 programas. Por mi parte, les haré una pregunta que me gustaría que contestaran en su revista:

¿No les parece excesivo el precio de los programas? Yo por ejemplo, no dispongo de mucho dinero y he ido a comprar programas al Rastro varias veces.

¿No sería mejor que las casas de programas bajaran los precios? Si así fuera, sería preferible comprar una cinta original que una copia pirata.

J. M. G. Madrid

☐ Si alguien roba su trabajo a los demás, es evidente que puede venderlo más barato. Si usted se pasara un mes entero escribiendo un programa, seguramente no le haría ninguna gracia ver cómo unos piratas se embolsan los beneficios de su trabajo. La piratería de software es como la venta de artículos robados, el vendedor puede poner un precio bajo porque no tiene gastos, todo son beneficios.

No obstante, a la larga, la piratería es perjudicial para todos porque, si las casas de software no venden, dejarán de escribir programas y entonces, ¿quién cree que los escribirá?, ¿los piratas? No se deje engañar por unos bajos precios, la mayoría de las copias piratas no funcionan, dan problemas de carga o no incluyen instrucciones y, en cualquier caso, a la larga le están defraudando también a usted.

El precio del sotfware para Spectrum no es tan caro si tenemos en cuenta que un programa para, por ejemplo, un IBM-PC cuesta alrededor de 200.000 pesetas. Tenga en cuenta que los programas no se hacen «a máquina»; para hacer un programa es necesario que un equipo de especialistas trabajen durante días, a veces, durante meses. Todo para que luego un pirata, que ni siquiera sabe programar, se apropie de los beneficios de su trabajo.

No es esta la única forma por la que nos perjudican los piratas. A todos nos gustaría que nuestro software no estuviera protegido para poder entrar en él y cambiarlo a nuestro antojo. Sin embargo, nos lo tienen que dar protegido por culpa de los piratas. Muchas veces, la protección de un programa lleva tanto tiempo como su elaboración, lo que incrementa, aún más, el precio. Si no existieran piratas, los programas se podrían vender a menos de la mitad de precio. ¿Cree todavía que los piratas le están haciendo un favor?

Los juegos de muchas pantallas

¿Cómo se las arreglan para codificar los gráficos y atributos que constituyen las 100, 200 e incluso más pantallas de algunos juegos comerciales?

José A. GIL - Madrid

☐ Si se almacena en memoria una pantalla octeto a octeto, ocupará 6K, por lo tanto, no será posible almacenar más de 2 ó 3. Ahora bien, las pantallas de los juegos comerciales suelen estar formadas por una combinación de bloques iguales. Imagínese que diseña una serie de gráficos de 16 × 16 pixels, como si fueran caracteres, pero más grandes. Combinando estos gráficos, pueden formar tantas pantallas como quiera, y cada una no ocuparía en memoria más que 16 × 12 = 192 bytes, con lo que podría almacenar muchas.

Esta es una de las técnicas más usadas para crear pantallas. En ocasiones, los bloques gráficos pueden ser más grandes, por ejemplo de 32 × 32, en este caso, cada pantalla ocuparía, tan sólo, 48 bytes.

"GP-50S" y "GENS-3"

Me gustaría que me contestaseis a varias preguntas referentes a la impresora «GP-50S» que es la que tengo.

—¿Es posible, por medio de alguna rutina en código máquina, hacer doble pasada?

—¿Se puede variar la velocidad de impresión?

—¿En el curso de código máquina vais a dar información del funcionamiento de los programas GENS y MONS? (Tengo las instrucciones, ¡pero en inglés!).

Iñigo LARIA - Madrid

☐ Si se diseña sus propias rutinas de impresión en lugar de utilizar las de la ROM, sería posible conseguir que la impresora hiciera dos pasadas. Sin embargo, no serviría para nada, ya que la letra de calidad se consigue haciendo las dos pasadas con los puntos ligeramente desplazados entre sí.

Respecto a la velocidad de impresión, viene dada por el propio motor de la impresora por lo que es imposible variarla.

Como ya dijimos al principio del curso, en único fallo del maravilloso ensamblador GENS-3 es que trae las instrucciones en inglés. Para subsanar este problema, dedicaremos un capítulo a explicar en profundidad la utilización de este programa.

"Línea 0"

¿Cómo se pone una línea 0?

Iñaki RODRIGUEZ

☐ La línea 0 debe ser la primera línea del programa. El procedimiento a seguir es el siguiente:

1. Escribir la línea poniéndola como número el «1».
2. Teclar el siguiente comando directo:

POKE PEEK 23635 + 256 *PEEK 23636 + 1,0

"VOYAGER"

El otro día tecleé el programa «VOYAGER», publicado en el número 69 y al ejecutarlo, me salió el mensaje: «C Nonsense in Basic 170:5». Comprobé una y otra vez el programa, pero nada. Se me ocurrió luego, cambiar el RESTORE de la línea 170 por RETURN y el programa salió bien, aunque los gráficos J, K, L, M, N, G, H y P no salieron, en su lugar, salían las letras. Volví a teclear esa línea correctamente, pero nada.

Victor GIMENO - Valencia

☐ La misión de la línea 170 es leer los datos que componen los gráficos y almacenarlos en su lugar correspondiente. Si anula esta línea colocando un RETURN al principio, los gráficos no se generarán.

El comando 5 de esta línea es «READ s». En principio, este comando no puede producir el error que nos indica, por lo que le re-

comendamos que compruebe esta línea y verifique si éste es realmente el quinto comando.

Por otro lado, el error puede estar también en los datos de las líneas 180, 190, 200 y 210 por lo que le recomendamos que se centre en revisar estas líneas. El resto del programa debe estar correcto desde el momento que funciona al anular los gráficos.

Código máquina

Respecto a las fichas de rutinas. ¿Se pueden incluir dichas rutinas en los programas que realicemos?

En un programa que estoy realizando, necesitaría borrar el programa Basic pero que la parte de código máquina siguiera funcionando. ¿Cómo podría hacerlo?

¿Hay alguna forma de situar una rutina en código máquina en la RAM de forma que se ejecute como si se tratara de una interrupción?

José BAU - Valencia

☐ Nuestras rutinas tienen una misión fundamentalmente didáctica, por lo que puede incluirlas en sus programas, siempre y cuando no lo haga con fines comerciales.

Si el código máquina está colocado por encima de RAMTOP, puede borrar el Basic con «NEW» y esto no afectará, para nada, al código máquina. Ahora bien, si lo que desea es borrar el Basic desde el programa en C/M, puede utilizar la subrutina de la ROM «RECLAIM-1» cuya dirección es 19E5h (6629). En esta rutina debe entrar con el registro «DE» conteniendo la primera dirección a borrar y el «HL» conteniendo la siguiente a la última. En nuestro caso, «DE» debe contener la dirección almacenada en «PROG» (23635) y «HL» la almacenada en «E-LINE» (23641), así que la rutina puede ser la siguiente:

```
LD DE, (23635)
LD HL, (23641)
CALL #19E5
```

Esta subrutina la hemos utilizado nosotros en el programa «CIRUJANO» publicado en el número 54.

Es perfectamente posible ejecutar una rutina por interrupción. El procedimiento consiste en seleccionar el MODO 2 de interrupción y direccionar la rutina con el vector de página del registro «I» y un vector de interrupción. El procedimiento es fácil, pero resulta muy largo de explicar aquí. No obstante, en nuestro CURSO DE CODIGO MAQUINA, se explicará en profundidad cuando tratemos las instrucciones de control de la CPU.

DE OCASION

● VENDO Wafadrive en perfecto estado, comprado en julio-85. Precio: 30.000 ptas. Interesados llamar al tel. (981) 78 29 52. José Manuel Cagigao c/ Loyos, 13. Miño. La Coruña.

● VENDO Spectrum 48K en perfecto estado, con cables, fuente de alimentación, libros, embalaje original. Todo por 15.000 ptas. Interesados llamar al tel. (91) 695 38 45. Preguntar por Andrés.

● VENDO vídeo-juego Atari en perfecto estado de funcionamiento con dos joysticks, adaptador TV, y también dos joysticks Paddle. Precio: 18.000 ptas. Te. (91) 273 24 20.

● VENDO Zx Interface con Microdrive, e instrucciones, todo nuevo, por 25.000 ptas. Interesados llamar al tel. (911) 42 72 06 y preguntar por Juan Carlos.

● VENDO vídeo-juego Phillips por 20.000 ptas. o bien lo cambio por impresora Seikosha GP-50S. Interesados escribir a la siguiente dirección: Luis José Macías. C/ Baños, 14, Mérida. Badajoz. Tel. 31 54 69.

● VENDO Zx Spectrum Plus, nuevo. Precio a convenir. Interesados llamar al tel. (94) 464 52 40. Preguntar por Jaime.

● VENDO por cambio de ordenador los fascículos del 1 al 60 de Microhobby con tapas para los 20 primeros números por sólo 4.800 ptas. También vendo 2 libros de Enciclopedia Práctica del Spectrum Run, encuadernados los dos primeros por 2.900 ptas. Estos precios pueden ser negociables. Interesados llamar al tel. 653 40 77 de Madrid. Julio Barquilla Rodríguez c/ Manuel Jiménez, 1, 2.° San Sebastián de los Reyes. Madrid.

● OFREZCO 300 ptas. más gastos de envío a la persona que me pueda facilitar la protección del Dragontorc. Interesados llamar al tel. (978) 41 66 79 o bien escribir a Regina Lanzas Blanco. Pseo. San Antonio, 11, 1.° D. León.

● VENDO Spectrum 16K en perfecto estado, con manual y libros en español. Tel. (91) 218 05 69. Preguntar por Juan José.

● VENDO ordenador Spectrum Plus, en perfectas condiciones, con cables, transformador e instrucciones en castellano. Urge. Llamar al tel. (986) 29 92 34 a partir de las 7 tarde. Preguntar por Juan Carlos.

● COMPRO instrucciones de programas para Spectrum que estén en español. También intercambio ideas, e información. Interesados escribir a la siguiente dirección: Evelio Gómez Pache. Sta. Eulalia, Grupo B-Bjo. Mérida. Badajoz. Tel. (924) 31 76 06.

● VENDO circuitos integrados (TTL, C.-Mos, micros, memorias) para ordenador y demás usos generales. Mando lista. Interesados dirigirse a Miguel Angel Clavijo. José Antonio, 5, 1.° Iz. La Cuesta. Sta. Cruz de Tenerife.

● CAMBIO un radio-cassette de coche (Phillips) stereo con ecualizador gráficos, leds sicodélicos, con amplificador de 40 volt. por canal, pudiéndose conectar al coche y/o casa. Valorado en 25.000 ptas. Lo cambio por una impresora GP-50S de Seikosha o similar (menos la Zx-Printer que no utiliza papel normal). Interesados pueden ponerse en contacto con Juan Muñoz. C/ Del Rebeco, 5, 3.° 1.ª. 08033 Barcelona.

● ESTOY interesadoo en formar un club sobre el Zx Spectrum. Interesados escribir a Carles Jordi Fernández. C/ Crta. a Bagá, 42. Guardiola de Berguedá. Barcelona.

● VENDO Spectrum 48K Plus (en garantía), libros y revistas. También vendo monitor de fósforo verde por 19.000 ptas. Interface monitor monocromo color «sinter» por 3.500 ptas. Junto o separado. Interesados llamar al tel: (93) 218 33 64. Emilio.

● CAMBIO Zx Spectrum con los siguientes accesorios: libros, 20 revistas de Microhobby, 15 de Zx, dos de Micromanía, 5 de Load In Run, un estuche de cintas. Lo cambio por un Amstrad o MSX 64K. Interesados escribir a la siguiente dirección: Juan José Granados Luque. C/ Santa Rosa, 49, 6.°. Alcoy. Alicante.

● VENDO Zx Spectrum 48K, en perfecto estado, íntegro, con todos los accesorios, cables, fuente de alimentación, cinta de demostración, manuales en castellano, embalaje original, etc. Además incluyo revista del primer año de Microhobby Semanal. Todo ello por 33.000 ptas. Interesados llamar al tel. 361 99 00 de Valencia y preguntar por Manolo (horario de oficinas).

● VENDO ordenador Spectrum Plus con todos los accesorios, TV en B/N de 12", nuevo, libros, revistas. También vendo calculadora programable Casio PB 110 con ampliación de memoria en 12.000 ptas. Interesados llamar al tel. 679 25 47. Madrid. Preguntar por Francisco Alvarez.

● VENDO Microhobby del n.° 1 a 42. Precio: 50 a 60 ptas. ejemplar. Fernando. Tel. (988) 72 46 79.

● VENDO Zx Spectrum 48K, con salida de vídeo, Zx Interface I, Microdrive, interface impresora parelelo Centronics, paquete ensamblador Devpac Gens-Mons, seis libros de Código Máquina, 9 libros de Basic y casi 100 revistas sobre el tema. Comprado en jun-85, con embalaje original e instrucciones. Precio a convenir. Interesados llamar al tel. (93) 351 65 26 de Barcelona. Preguntar por Joaquín.

● VENDO dos plantillas para Spectrum 16 ó 48K, junto con los manuales del mismo en español, la cinta Horizontes y la del Spectrum Plus de demostración, todo por 3.600 ptas. Acepto otras ofertas. Interesados escribir a la siguiente dirección. Andrés Felipe Hernández López. Pza. de la Constitución, 14, 3.° Molina de Segura. Murcia. Tel. (968) 61 18 39 (1 a 3 y 5a 9).

● VENDO Zx-81, casi sin usar, con todos los complementos, todo en perfectas condiciones. Precio a convenir. Llamar por las tardes al tel. (948) 24 29 82. Pamplona. Navarra. Preguntar por Tomás.

● VENDO consola de vídeo-juegos Phillips G-7000 en perfecto estado, por 20.000 ptas. Interesados llamar al tel. 446 87 88. Preguntar por Alberto. Madrid.

● DESEARIA vender Spectrum 48K, cableado completo, estabilizador especial, más 50 revistas, 28 fascículos de «Run». Todo por 23.000 ptas. Tel. (91) 439 83 18. Preguntar por Marcos.

● VENDO Spectrum 48K en buen estado con fuente de alimentación, manual de instrucciones y cinta Horizontes en 25.000 ptas. José Antonio Pérez Moreno. Hermanos Sánchez del Río, 1, C-3. Gijón. Tel. (985) 33 62 67.


ACCION!
SAI COMBAT
¡EL DEFINITIVO!
— 8 niveles de dificultad desde cinturón blanco a cinturón negro.
— A partir del octavo Dan puedes llegar a ser "Sai Master".
— 16 Movimientos diferentes.
— Joystick o Teclado.
Spectrum Pronto: Commodore
Amstrad
Si están agotados en tu tienda habitual ¡¡LLAMANOS!!
ACCION!
Spitfire
El mejor programa de simulación de vuelo creado hasta la fecha.
Pon en marcha los motores... ¡Despega!
Sube al cielo para defender el país.
Derriba a tu enemigo para ganar condecoraciones y ser uno de los pilotos de élite de la R.A.F.
Commodore 48K
Spectrum 64
Editado, fabricado y distribuido en España bajo la garantía Zafiro. Todos los derechos reservados
MIRRORSOFT
ONLY THE BEST
ZAFIRO SOFTWARE DIVISION. Paseo de la Castellana, 141. 28046 Madrid. Tel. 459 30 04. Tel. Barna. 209 33 65. Telex: 22690 ZAFIR E

SINCLAIR STORE
EL CENTRO DEL HARDWARE

SPECTRUM 48 K
SPECTRUM PLUS
SPECTRUM 128
SINCLAIR QL
COMMODORE 64
COMMODORE 128
COMMODORE PC 10
COMMODORE PC 20
AMSTRAD 472
AMSTRAD 6128
AMSTRAD 8256
Y ....
SPECTRAVIDEO
MSX



- EN SINCLAIR STORE USTED NO PAGA EL **IVA**
- POR LA COMPRA DE UN ORDENADOR, CURSO **GRATIS** DE INFORMATICA
- EN TODAS LAS IMPRESORAS 20% DESCUENTO
- JOYSTICK QUICK SHOT II + INTERFACE TIPO KEMPSTON **3.800 Pts.**
- JOYSTICK ANATOMICO AMARILLO + INTERFACE TIPO KEMPSTON **3.200 Pts.**
- PC COMPATIBLE IBM P.V.P. 212.000 Pts.
- COMPRA 3 JUEGOS Y PAGA SOLO 2. LOS MEJORES JUEGOS DESDE **990 Pts.**
- 4 CARTUCHOS MICRO DRIVE CON CARTUCHERA **2.200 Pts.**
- PRECIOS ESPECIALES PARA COLECTIVOS Y EMPRESAS
- DISTRIBUIDORES OFICIALES DE TODAS LAS MARCAS. CON AUTENTICO SERVICIO PROFESIONAL DE POST-VENTA
- VEN A VERNOS, NOSOTROS MANTENEMOS LAS **REBAJAS,** EN TODOS LOS ARTICULOS.
- NECESITAMOS DISTRIBUIDORES. SOMOS MAYORISTAS
- SERVICIO TECNICO DE REPARACIONES **3.700 Pts.**
- ULTIMAS NOVEDADES PARA AMSTRAD: AMPLIACION DE MEMORIA 64 K, 256 K, y SILICON DISK
- ULTIMA NOVEDAD EN EL MERCADO ATARI 520 S.T. Y ATARI 1.040 S.T. YA DISPONIBLE

¡Ven a probarlos!

sinclair store

SOMOS PROFESIONALES

BRAVO MURILLO, 2
(Glorieta de Quevedo)
Tel. 446 62 31 - 28015 MADRID
Aparcamiento **GRATUITO** Magallanes, 1

DIEGO DE LEON, 25
(Esq. Nuñez de Balboa)
Tel. 261 88 01 - 28006 MADRID
Aparcamiento **GRATUITO** Nuñez de Balboa, 114

FELIPE II, 12
(Metro Goya)
Tel. 431 32 33 - 28 009 MADRID
Aparcamiento **GRATUITO** Felipe II